Scribbling Speech Using Urdu

Submitted by

Muhammad Hasnain Khan
19i-1255

Supervised by

Dr. Mirza Omer Beg
Masters of Science (Computer Science)

A thesis submitted in partial fulfillment of the requirements for the degree of
Masters of Science (Computer Science)
at National University of Computer & Emerging Sciences

Department of Computer Science
National University of Computer & Emerging Sciences

Islamabad, Pakistan.

February 2021

Plagiarism Undertaking

I take full responsibility of the research work conducted during the Masters Thesis titled *Scribbling Speech Using Urdu* . I solemnly declare that the research work presented in the thesis is done solely by me with no significant help from any other person; however, small help wherever taken is duly acknowledged. I have also written the complete thesis by myself. Moreover, I have not presented this thesis (or substantially similar research work) or any part of the thesis previously to any other degree awarding institution within Pakistan or abroad.

I understand that the management of National University of Computer and Emerging Sciences has a zero tolerance policy towards plagiarism. Therefore, I as an author of the above-mentioned thesis, solemnly declare that no portion of my thesis has been plagiarized and any material used in the thesis from other sources is properly referenced. Moreover, the thesis does not contain any literal citing of more than 70 words (total) even by giving a reference unless I have the written permission of the publisher to do so. Furthermore, the work presented in the thesis is my own original work and I have positively cited the related work of the other researchers by clearly differentiating my work from their relevant work.

I further understand that if I am found guilty of any form of plagiarism in my thesis work even after my graduation, the University reserves the right to revoke my Masters degree. Moreover, the University will also have the right to publish my name on its website that keeps a record of the students who plagiarized in their thesis work.

____

Muhammad Hasnain Khan

Date: ____________________

Author's Declaration

I, Muhammad Hasnain Khan, hereby state that my Masters thesis titled *Scribbling Speech Using Urdu* is my own work and it has not been previously submitted by me for taking partial or full credit for the award of any degree at this University or anywhere else in the world. If my statement is found to be incorrect, at any time even after my graduation, the University has the right to revoke my Masters degree.

____

Muhammad Hasnain Khan

Date: ____________________

Certificate of Approval

It is certified that the research work presented in this thesis, entitled "Scribbling Speech Using Urdu " was conducted by Muhammad Hasnain Khan under the supervision of Dr. Mirza Omer Beg.

No part of this thesis has been submitted anywhere else for any other degree.

This thesis is submitted to the Department of Computer Science in partial fulfillment of the requirements for the degree of Masters of Sciencein Computer Science

at the

National University of Computer and Emerging Sciences, Islamabad, Pakistan

February' 2021

Candidate Name: Muhammad Hasnain Khan Signature: ____________________

Examination Committee:

1. Name: Dr. Noreen Jamil Signature: ____________________
   Associate Professor, FAST-NU Islamabad.

2. Name: Dr. Mehreen Alam Signature: ____________________
   Assistant Professor, FAST-NU Islamabad

Dr. Hammad Majeed ______________________________
Graduate Program Coordinator, National University of Computer and Emerging Sciences, Islamabad, Pakistan.

Dr. Waseem Shahzad ______________________________
Head of the Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan.

Abstract

Languages and images are closely related. We explain facts as the spatial constellation and we think in pictures. What if the spoken sentence could be transformed into the visual worlds in real-time? Machine learning, speech analysis, and recurrent neural networks for image generation allow a computer to understand the natural language and also generate a complex imaginary world following the user speech and thus create complex animations controlled by linguistic structures. Generating a story or scene from voice is a unique problem that got a lot of attention in the past few years but to the best of our knowledge literature in this field is very much limited. In this research work, we working on making machines very intelligent so that they can interpret the natural language in such a way that some entities will be extracted from a sentence with context such as one entity is ”above or below” or ”before or after” to another entity. After extracting entities then some generative networks will be used to generate the identified objects according to the context. Our system will be able to generate real-time 3D models to make it more interactive for the user. We are proposing a module approach to solve this problem. We will work on three modules; first module will be a machine learning model to extract the entities from the speech input. Second module will be built for extracting the relations of the identified entities. Third module will be able to generate 3D models of the identified entities with respect to their relation extracted from second module. For speech input, we will be using *Google ASR*.

Acknowledgements

I would like to thank all the little people who made this possible.

Dedication

This is dedicated to the one I love.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Humen interact with each other in many ways like facial expression, gesture and particularly speech. Image has great influence on us. All images that we see are influencing us in a way all the times. Our brain absorbs all the visual information subconsciously. We respond to and process visual data better than any other types of data. According to a study [1] visuals like images and videos are processed 60,000 faster than text. Just like the mail or telephone, image is a medium which carries a message. To solve the scribbling speech using Urdu, we will have to know the working of speech to text conversion and then tokenize the text using NER. The basic flow of speech recognition for urdu includes three main steps, conversion of an analog signal(waveform) to digital representation is the first step. Second step is to break down the digital represented waveform into some distinct and unique units of sound also known as pause and phonemes. Third step is to run all the distinct units of sound through an algorithm to determine the resulting text [1]. But how does algorithm works in the third step to formulate text from phonemes? All speech recognition systems have to have a dictionary which consists of a list of words and then a recognizer will identify the combination of phonemes that makes a word [2]. There are a few challenges in recognizing phonemes like it is possible for a single word to have many phonemes combinations as a word can have multiple definitions in a language dictionary [3]. There are two types of speech recognition systems local and remote [4]. We will be using remote speech recognition system that means dictionary file will be stored on the device where speech recognition algorithm is running [4]. To be precise, we will be using **Google Speech Recognition** API for ASR.

Learning to generate story from the audio is a challenging task and it is only done for English language which is a high resource language but this problem becomes more challenging when it comes to low resource language like Urdu. There are multiple modules that needs to be done to accomplish this task; first module is to convert the Urdu speech into text which is a tedious task. Second module is to extract the meaningful information from the text using NER which will extract the entities and every detected entity will be classified into a predetermined category.For example,

[1] http://www.t-sciences.com/news/humans-process-visual-data-better

an NER machine learning (ML) model might detect the word “super.AI” in a text and classify it as a “Company”. Third and final module will be used to draw the detected entities according to the speech. With NLP techniques like NER it is possible to extract nouns, adpositions like ”above”, ”under” and so on from the real time streaming. RecognizeStream, getEntities and analyzeSyntax are used to realize the input part. For example, we give this sentence سمندر کے اوپر پرندا اڑ رہا تھا then our system should be able to extract entities like سمندر, اوپر, پرندا and اڑ. Scribbling speech is dependent on multiple modules and the most important module is to extract the entities from the voice and the visuals in scribbling speech will be more clear when the extracted entities are extracted in a meaningful way. Story generation with image captioning and GAN has been done before but to know the importance of the story generation from text without we will have to know how Generative Adversarial Network (GAN) works. Generative modelling is an unsupervised learning task where machine automatically discovers patterns and learns the regularities in that patterns. GAN usually has two main components [5]. First component is generator that learns to generate data. Second component is the discriminator which classify the generated data as real or fake. These two components are trained together until the discriminator model is fooled which means that generator model is generating real examples. Scribbling Speech is a new field and till now only Google has worked on it. There is a room of improvement as it is a new direction for researchers and it will open door of research for many researchers. It will be beneficial for people to work in new environment with the help of available research content which is still lacking in this era of technology about this certain specific area of research. Urdu is spoken by more then 70 million people in the world and most people are from Pakistan or India. More then 50 languages are spoken in Pakistan including Urdu which is the national language of Pakistan. Urdu is written from right to left in Nastalique style and Urdu is an Indo-Aryan language [6]. Urdu is written in cursive Arabic script that’s why there are no spaces between the words and most of the words are compound word. In Urdu, characters are joined together to make a full word and usually one character can achieve different shapes to complete a word. These characters having different shapes are called joiners. Urdu is somewhat related to Hindi as both of the languages are originated in subcontinent India, both languages belong to Indo-Aryan and they are very familiar in phonology and grammar. Also there are other languages which follow right to left convention like Arabic, Farsi, Punjabi, Pashto etc. These languages are a little bit common as they share the script and some vocabulary that’s why a language specific task can only be used for that language because vocabulary is different for all of the right to left languages. Urdu share it’s resources with Hindi and the key difference between these two language is the writing style as Urdu is written in Arabic style whereas Hindi is written in Devangari style. Urdu and Hindi are almost similar languages but Hindi language processing techniques can’t be applied on Urdu [7]. Urdu is quite complicated language because Urdu language has a characteristic of embedding lexical features from other languages particularly from English. This phenomena is known as **Code Switching**. For example there can be sentence in Urdu where the sentence start from right to left but somewhere in the text there is a English word which is left to right. For example a whole sentence flow can be interrupted by inserting a single word of English i.e جملے کے درمیان میں الفاظ Urdu. But there are compilers which works with Urdu and if you compile the above sentence in Latex Compiler then it will recognize that there is an English word embedded in the Urdu sentence and it will change the sentence like this. جملے کے درمیان میں اردو

1.1 Named Entity Recognition (NER)

Named Entity Recognition is a tedious task in NLP. NER is an important task in many natural language processing applications like machine translation, information extraction etc. NER is a challenging task and a huge amount of literature can be found on NER for English and other Western languages. Plenty of resources can be found in the western languages. Named Entity Recognition is a crucial task because of the non-availability of the resources and most of the text prepossessing techniques don't work on Urdu because of its morphological complexity. Words in Urdu are not like English words, in Urdu most of the words are compound words that's why words segmentation is also a challenging task here. In this paper, we present the development of NER tagged dataset for NER research in Urdu and and a neural network architecture which includes bidirectional LSTM with CNN and CRF layer which provides us the state-of-theart results. We have used FASTTEXT words embedding for Urdu which are available publicly. Our NER dataset contains about 54,000 words which includes 7 different classes. A few papers can be found on NER for Urdu and different techniques have been used such as Rule Based Learning, hidden Markov model (HMM), RNN etc. In this section, firstly, we will discuss the about the NER history then we will briefly explain the Urdu Language like how it is different from other Arabic script language and what are the problems we could face in the pre-processing of Urdu text. After that we will discuss the challenges that will be faced for Urdu NER. Available datasets, related work and our implementation will be discussed later in this section.

1.2 NER Challenges

NER is a difficult task in NLP and this task is more challenging when it comes to scarce resource languages like Urdu. In the above section, we have explained the Urdu grammar rules and writing style and in this section we will address all the problems we faced in Urdu for NER problem.

1.2.1 Agglutinative Nature

In 2008, **IJCNLP** described that a word and it's meaning can be changed by adding just a single word which is known as Suffix. **IJCNLP** mentioned this feature for Telugu language but agglutinative nature can also be applied on Urdu and most of the agglutinative nature of Urdu comes from the Turkish, Persian and Dravidian languages [8]. By adding a single word to the root form of a word can not be recognized as named entity. For example, in Urdu a word Pakistan + i = Pakistani will not be recognized as named entity and Pakistan is a location named entity.

1.2.2 Different Variants of Spelling

There can be found different spelling of a word in Urdu for example a city named راولپنڈی also written as راول پنڈی and this problem can be tackled very easily in English because of the capitalization feature but in Urdu there is nothing like capitalization.

1.2.3 Compound Words

Urdu has a characteristic of compound words where a few words can be concatenated to form up a new word. For example, a word اسلام آباد is a compound word of two words اسلام and آباد.

1.2.4 Words Ambiguity

A word can have different meaning in Urdu for example, word قمر is known for Moon and it is a name also. We can find plenty of these words in Urdu vocabulary and some words have more then 2 meanings. That's why it is very difficult to differentiate that a word is pointing to that direction without knowing the whole sentence.

1.2.5 Words from other language

Urdu has adopted multiple words from different languages like English, Persian etc. These words are called **Loan Words**. For example, we write *Umar Bin Al-Khattab* will be written in Urdu as عمر بن الخطاب. In above example *Bin*/بن is a loan word which is taken from Arabic vocabulary. Heaven/جنت is a loan word taken from Arabic. Desire/آرزو is a loan word in Urdu taken from Turkish.

1.2.6 No Capitalization

In languages like English, capitalization is the most important feature that can be used for NER. In English, capitalization can be used to recognize whether an identified entity is proper noun or not and many systems can easily recognize acronyms by looking at capitalization. But in Urdu there is nothing like capitalization. For example, in Urdu BBC/بی بی سی can not be recognized as acronym because of no capitalization.

1.2.7 Suffix Ambiguity

Suffix are meaningful word endings as it represents some useful information. It is very common in Asian languages to use location at the end of the name as suffix. It is a common practice in

Asian languages particularly Urdu to append the location of person's origin in a name with a suffix. For example if a person is from Lahore/لاہور (city is Pakistan) then adding suffix *i* at the end of person's name will represent that this person is from which city. Adding suffix *i* will end up adding *Lahori/لاہوری*.In Urdu, adding suffix with a name is very common for poets and singers. Adding a suffix with a name is like as alias and if the person is poet then the added alias will be called *takhalus/تخلص* .

1.2.8 Orders of Words

Most of the languages have word order and a few languages can be found which can be referred as free word-order. Most Asian languages have word-order different from English. Urdu has a word-order but word-order in Urdu depends upon the context and domain in which a sentence is being said. For example, میں نے ان سے دفتر میں ملاقات کی and میں نے دفتر میں ان سے ملاقات کی both of the above sentence meaning are same as ”I met him in the office”.

1.2.9 Resource Challenges

There are two main approaches that can be used for solving NER problems; one is *Rule Based* and other is *Machine Learning* techniques. For rule-base, it is necessary to know the in-depth details and complexities of Urdu language to make such rules which will be helpful in identifying named entities. Other approach is machine learning and machine learning techniques can get pretty good results for NER but to get good results, we will have to train a machine learning model on a large corpus. But unfortunately, due to unavailability of Urdu resources, this task is quite challenging. A large annotated corpus is the pre-requisite to make a good machine learning model.

1.3 Co-reference Resolution

Co-reference resolution is one of the important task in NLP. Resolving noun phrases to identified entities is known as co-reference resolution or entity resolution. Entity resolution has been divided in two sub areas known as **anaphora resolution** and **co-reference resolution**. Entity resolution plays an important role in man NLP applications such as sentiment analysis, machine translation, paraphrase detection, etc.

1.3.1 Anaphora Resolution

Anaphora is the use of an expression whose interpretation depends upon another expression in content. When we talk about anaphora that means we are referring backward to the pronoun which is related to another noun phrase. When the pronoun comes after the actual noun then that relation in the sentence is called an anaphora. For example, *The music was so loud that it couldn't be enjoyed.* In this sentence ”The music” is noun and in the sentence ”it” is referring to the noun which was mentioned in the start.

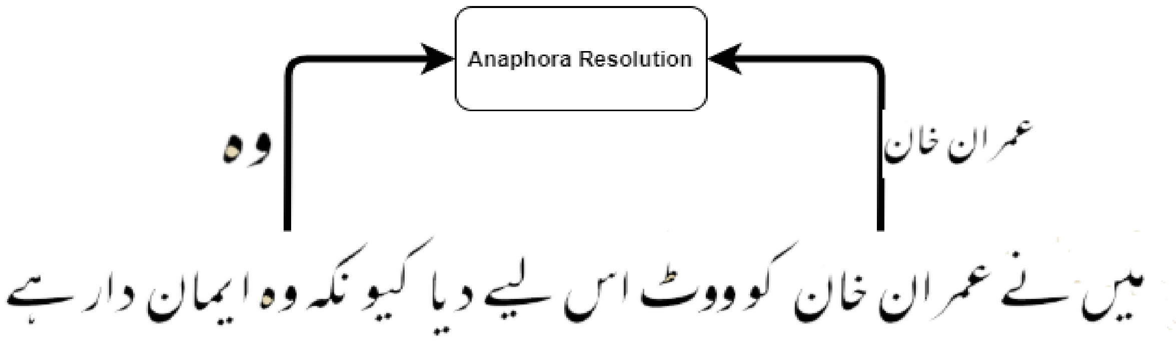


Figure 1.1: Anaphora Diagram

1.3.2 Cataphora Resolution

Cataphora is the other way around of anaphora. The pronoun comes before the actual noun it is referring to in the sentence. For example, *Despite her difficulty, Amina went ahead to help her.* In this sentence, the pronoun *her* is referring to the actual noun *Amina* and pronoun comes before the noun. This type of relation is called cataphora resolution.

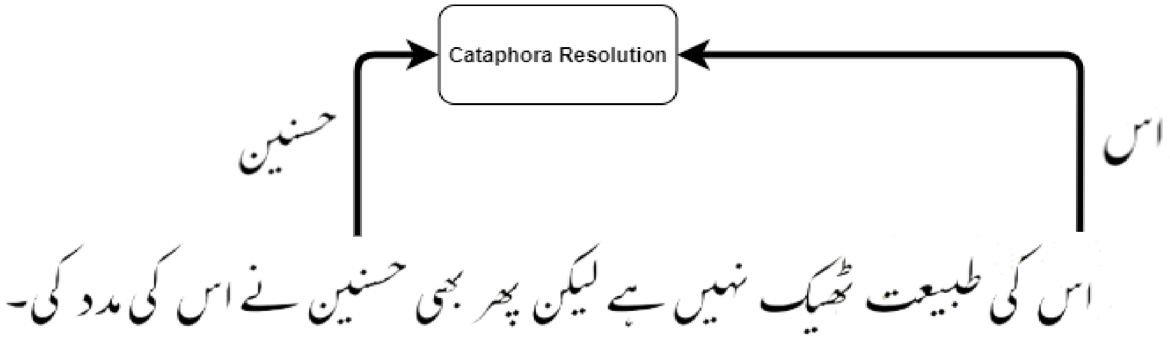

Cataphora Resolution
حسنین
اس
اس کی طبیعت ٹھیک نہیں ہے لیکن پھر بھی حسنین نے اس کی مدد کی۔

Figure 1.2: Cataphora Diagram

1.3.3 Anaphora vs Coreference

Coreference is referring to a noun without any linkage such as if we see *Obama* or *Barack Obama* in a sentence that means both words are referring to a particular noun. There can be many possible coreferences for a single noun without any linkage or dependence. On the other hand, Anaphora is linkage problem where one pronoun is referring to a noun and that noun will be referring to another noun that means a pronoun can not refer to a specific noun without creating an linker. For example, in a sentence *Barack Obama* is referring to the US president and then *He* is referring to the *Barack Obama* not the US president.

1.3.4 Zero Anaphora vs One Anaphora

Zero anaphora was first introduced in 1986 and it uses a gap between a clause or a phrase to refer back to its antecedent. One anaphora is exact opposite of zero anaphora and in one anaphora the word ”one” is referring to the antecedent. One anaphora is not very common in the language but it got much attention from the researchers in 2002 to solve it with machine learning approaches.

Our major contributions in this research work are given below:

- First, We created a NER dataset of 53K words having 8259 named entities.
- Second, We used a number of pre-trained words embedding for feature extraction.
- Third, some baseline models are created for NER with simple machine learning techniques.
- Fourth, we did experiments with different backbone architectures such as LSTM, BERT, Bi-LSTM with different necks such as CRF layer and self-attention layer to train a model for NER. Moreover, we used transfer learning to train NER model for Urdu from state-of-the-art models and Bi-LSTM and BERT outperforms the baseline models.
- Fifth, we created a dataset for co-reference of 2k sentences and created baseline model with 1k sentences.
- Sixth, We trained model with BERT for co-reference resolution and BERT outperforms baseline model.

Chapter 2

Literature Review

2.1 NER Related Work

NER is a task to identify and classify the identified nouns in their specific categories like person, organization, location, date, number, time, designation etc. NER has enormous number of applications such as information extraction, question answering systems, co-referencing etc [9]. A lot of work can be found in the literature on NER for English and other Western languages. Research work on NER for Urdu is still in early stages. The main reason behind the lack of literature of NER in Urdu is the non-availability of resources. There are multiple techniques that can be used for NER problem like supervised learning technique which include HMM, Rule-based learning method etc. NER is an important problem in the field of Natural Language Processing and it caught a lot of attention in the previous years. In 1990, Message Understanding Conference was doing research on information extraction and they realized that there is a problem which needs to be solved first to get the accurate information. That problem was named as NER (Named Entity Recognition). MUC worked on NER for the first time in the history. In literature mostly three approaches can be found for NER; Rule-Based, Machine Learning (SVM, HMM, KNN, Decision Tree) and Hybrid. approach. Rule-Based is always difficult to implement as we have to not only know the language but grammar rules also. NER systems for Urdu is in the developing state because most of NER systems rely on the plethora of resources. IJCNLP-08 is an NLP workshop and their main focus is on the development of systems for five languages i.e. Urdu, Bengali, Oriya, Telugu and Hindi. NER system for Urdu has a great potential because there are 70 million people who speaks Urdu. But because of the unavailability of resources, characteristics of Urdu makes this task more difficult. For other languages, capitalization has great importance in English and other languages for NER task but Urdu does not have capitalization. Rule Based Learning technique was famous in the start of the NER research and in 1991, a paper was presented was based on Rule Based system for NER identification and classification of different company names [10]. They achieved state-of-the-art accuracy in that time and accuracy was over 95%. In 1999, a language independent system was developed for NER for multiple languages like English, Turkish, Hindi and Greek [8]. Overall performance of this system was quite good but on Hindi the system performed very poorly and system has f-measure of 41.7 low recall 27.84% and precision 84%. [11] presented Maximum Entropy based NER system for English and has achieved 84.22% F-measure. Conditional Random Field has been used for the development of NER system for Hindi language with 71.5% accuracy [4]. For English NER a semi-supervised approach was presented for the development of English NER and in the paper they have classified 100 names entities and has achieved 83% F-measure [1]. An information extraction system was built for Urdu and two techniques were used for NER known as ME and CRF. F-measure of ME and CRF are 55% and 69% respectively [12].

2.2 Coreference Related Work

Hobbs Algorithm [13] was one of the earliest technique used to solve conference resolution problem. The algorithm was based on an abstract syntax tree with some specific rules. In late 90's machine learning research was started for solving coreference resolution problem. In 1998, to solve anaphora resolution problem a statistical approach was introduced [14]. Naive Bayes probabilistic approach was used first time for solving NLP problems starting with the coreference resolution [2]. Coreference resolution can be divided into a pairwise classification problem and decision tree approach was presented to solve this problem. Maximum entropy was used to train a classifier for determining if a noun phrase is anaphoric or not [15]. A new concept of *discourse parsing* was introduced which says that co-reference and anaphoric entities together make a subset of *discourse parsing* [16]. Coreference and anaphora resolution has been used in various applications such as sentiment analysis [17], text summarization [18], machine translation [19], question answering [5]. Entity cluster is an important problem and in 2016, it was proposed that coreference task can get benefit from this [20]. In the start, rule based learning was used in order to solve coreference problem but with the passage of time researchers have been active in this research area and and now they are using deep learning approaches to solve this problem. Recently bi-LSTM have been used to encode sentences and mention scoring and it is very important to know that how word and character level embedding can be combines with bi-LSTM, span head and span representation to get the best mention score [20]. In 2018, authors of the aforementioned paper gave an update about how the above trained model can be used antecedent scoring. They proposed that with the help of span representation and mention score we can get the antecedent score and coreference score. But one problem of the above proposed approach is that it is very difficult to maintain and the system is required to some domain specific adaption.

2.3 Story Generation Related Work

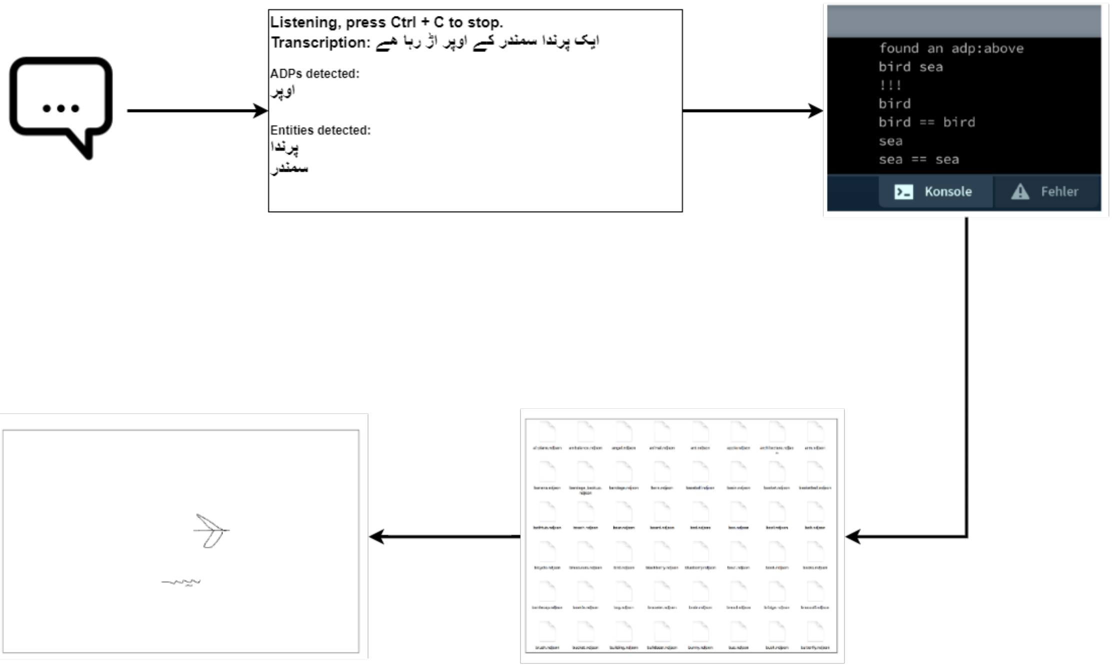
Generative models are considered Generative Adversarial Networks (GAN) have been used to create images from a text sentence and after StackGAN paper which was released in 2017, an immense amount of work being done using StacKGAN which is most similar to Conditional GAN. The StackGAN first outputs an image of resolution 64 and then takes this as prior information to generate an image of resolution 256. StackGAN has been used for image captioning and story generating from text. StackGAN is a very unique technique that takes text as input and generates an image according to text. This technique is unique because this is done using text embedding such that it captures the visual patterns in the environment. But the problem is that StackGAN is computationally expensive. And to generate a story from text is immensely expensive. So what if we can visualize the story from the speech without GAN. Most of the work done in story visualization using natural language Processing primarily focuses on textual data and most of the work done in this field of research is in English. But till now, no one used low resource language like Urdu to draw the visualization from speech in a virtual environment. Speech to text systems are being used widely in many applications. Speech to text for Urdu is an important part of this problem. Speech recognition can be referred as the process of converting analog signals to a sequence of words. Mel Frequency Cepstral Coefficients (MFCC) method is used which is based on the characteristics of the person human ear's hearing and then to simulate the person auditory system a nonlinear frequency unit is used. Main aim of story generation is to generate and visualise the story in such a way that that sequence of key static frames show the accurate and clear results. In 2019, StoryGAN has been presented which uses a recurrent neural network to cope up with the afore generated images into the image generation given a multi-sequence paragraph [21]. Reinforcement learning models or text generation models can be used also for visual storytelling [22]. Bi-directional multi-thread recurrent neural network are proposed in 2016 for story telling with the help of photo stream [22].

Paper Title	NER	CoReference	Approach	Feature Extraction	Dataset	Language
IJCNLP-2008	✓	✖	Hobbs	Rule Based	40,000	Urdu
Waqas et al	✓	✖	HMM	TF-IDF	32,000	Urdu
UNER	✓	✖	KNN	CountVec	48,673	Urdu
SpanBERT	✖	✓	BERT	Span-head	OntoNotes	English
CorefQA + SpanBERT	✖	✓	BERT + CorefQA	Global Features	CoNLL 2012	English

Table 2.1: Paper Comparison Table

Chapter 3

Problem Statement



```
Listening, press Ctrl + C to stop.
Transcription: ایک پرندا سمندر کے اوپر اڑ رہا ھے
ADPs detected:
اوپر
Entities detected:
پرندا
سمندر
```

```
found an adp:above
bird sea
!!!
bird
bird == bird
sea
sea == sea
```

Figure 3.1: Problem Statement Diagram

Languages and images are closely related and we could tell a story in our natural language and computer would be able to understand the story and transform the story in a dynamic virtual world where a camera will be available to give a realistic view. Machine learning, speech analysis and recurrent neural networks for image generation allow us to make an intelligent computer which will be able to generate complex imaginary worlds following the speaker and create complex structure and animations according to the context.

3.0.1 Research Contributions

To solve this problem, we endeavour to answer to following research questions;

- How would the dataset be manually annotated for NER in such a way that data meets the universal standards?
- Which vector embedding technique is most effective when it come to NER for Urdu language.
- How NER and co-reference resolution models can be integrated to solve a bigger problem?
- Collecting and annotating data for co-reference manually is a tedious task so what are the alternatives which can be used for data collection?
- How transfer learning can be used for training model for Urdu language?
- How NER and co-reference can be used to draw the entities on the screen according to the context?

Chapter 4

Methodology

We are proposing a module approach where three different modules will be built to work together to solve the scribbling speech problem. In this chapter, we are going to give the detailed analysis of datasets and how different modules will be implemented.

4.1 NER Available Datasets

NER problem can be solved with supervised and unsupervised techniques and in supervised machine learning technique, we will have to have a huge corpus which is not available for Urdu. Till now, there are three datasets which are available online and these datasets are free to use. First dataset in *IJCNLP-2008* NE tagged annotated dataset which can be used with supervised machine learning. Details of **IJCNLP-2008** dataset are given below.

Named Entity	**IJCNLP-2008**
Person	277
Organization	490
Location	48
Date	123
Number	108
Designation	69
Total	**1115**

Table 4.1: IJCNLP-2008 Dataset

IJCNLP-2008 dataset consists of 40,000 total words which contains 1115 named entities.

Second NER dataset consists of 32,000 words and it contains 1526 named entities and the dataset is annotated for four named entities. Details of the second NER dataset can be found below.

Named Entity	Waqas,Jahangir et al
Person	380
Organization	756
Location	282
Date	101
Total	**1519**

Table 4.2: Waqas et al Dataset

Researchers at Islamic International University complied a dataset which contains 48,673 words with 4621 named entities and this dataset is annotated for 7 different classes. This dataset was collected from national news, international news and sports news and the dataset is available online. This dataset is known as UNER.

Named Entity	UNER
Person	1207
Organization	633
Location	1205
Date	203
Number	991
Designation	279
Time	73
Total	**4591**

Table 4.3: UNER Dataset

In **UNER** dataset, out of 4621 named entities, entities from sports news are 1809, entities from international news are 1088 and entities from national news are 1749.

4.2 Co-Reference Resolution Proposed Approach

In this section, we are going to explain the implementation of Co-Reference Resolution.

4.2.1 Dataset Collection

Same procedure is followed to collect the co-reference dataset as we used for NER dataset. We collected dataset from BBC Urdu News and other Pakistani national news channels. We collected 1,000 news from different sources.

4.2.2 Preprocessing

The collected dataset is noisy so basic preprocessing techniques are applied to remove the noise and make the text clean. After cleaning the text, dataset was annotated in anaphora and cataphora resolution format as famous dataset of co-reference *GAP* did.

4.2.3 Evaluation Metrics

A number of metrics are available for the evaluation of co-reference resolution. Some standard evaluation metrics are explained below.

MUC-Link based F-measure

This evaluation metric was introduced in MUC 6th conference and this evaluation metric considers a cluster of references as linked references and each reference is linked to more than 2 other references.

Constrained Entity Alignment F-measure

This metric is used to evaluate the entity-based similarity identification. Similarity measures are used to create an optional mapping between predicted clusters and truth clusters.

Bagga and Baldwin's B-cubed metric

This metric was proposed in 1998 and it considers each individual reference to compute the precision and recall and takes weighted sum of computed precision and recall.

CoNLL Score

This score considers the average of the B-cubed score, MUC score and the CEAF score.

There are other evaluation metrics which can be seen in fig 4.1, evaluation metric image is taken from [1].

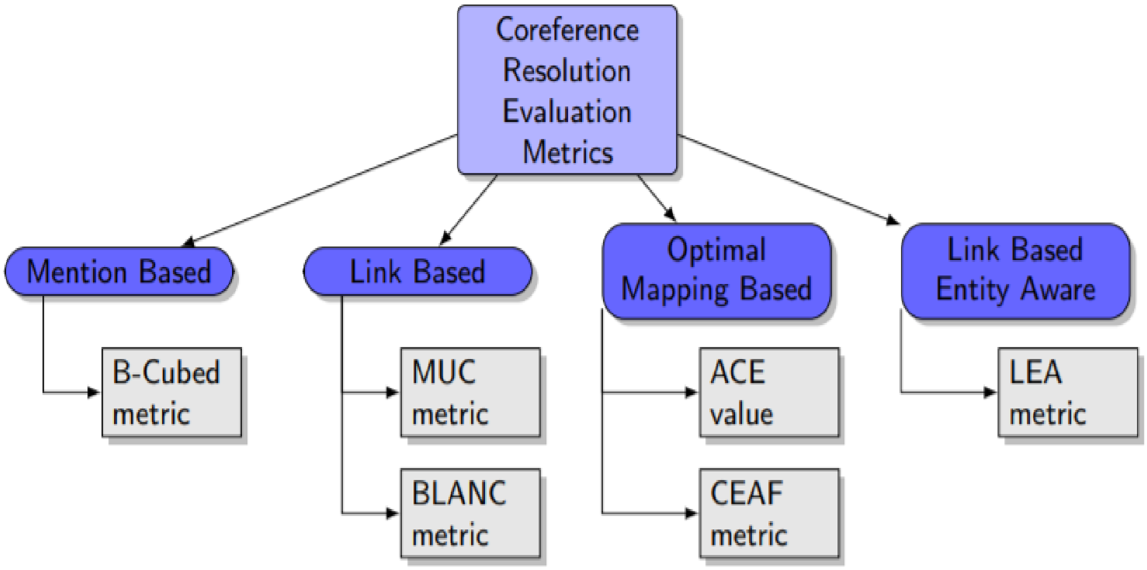


Figure 4.1: Co-reference Evaluation

4.2.4 Co-Reference Resolution Evaluation and Experiment

For co-reference resolution, we used *Ktrain* library as a wrapper for Urdu in BERT-base and BERT-large. We used pre-trained BERT model and feed forward network. For evaluation, we used *MUC*, *B-cubed* and *CEAF* evaluation metric.

From the table 4.4, it can be seen that BERT-Large is giving almost 50% accuracy with 2k sentences. Performance can be improved by increasing the dataset as same BERT-large model give 83.5% F1 measure with *GAP* dataset.

[1] https://www.semanticscholar.org/paper/Anaphora-and-Coreference-Resolution

Model	Metric	Precision	Recall	F1 Measure
BERT-base	MUC	48%	42.3%	46.2%
BERT-base	B3	39%	39.7%	40.2%
BERT-base	CEAF	39.3%	37.4%	38.1%
BERT-large	MUC	50.3%	43.6%	48.4%

Table 4.4: Co-Reference Results Comparison Table

4.3 NER Experimental Setup

In this section, we explained all the details of experimental setup which was used to get the results for different embedding and various models.

As we have discussed that in the literature various feature extraction techniques are used and a few techniques can be found specific to Urdu language. Therefore, we have also used various embedding techniques to retrieve a feature vector against each named entity and sentence in case of co-reference. We used simple machine learning techniques such as *TF-IDF* and *CountVectrizer* along with *KNN* and *SVM* to create baseline models.

4.3.1 Baseline Models

For the purpose of acquiring some baseline benchmark results on the dataset, we have trained three different baseline models with all the dataset we had.

TF-IDF + KNN: K-Nearest Neighbours is one of the most common choice when it comes to classification problem. We have used TF-IDF vector representation to get words embedding which will be used by KNN for classification.

TF-IDF + SVM: Support Vector Machines are the most appropriate choice form the most common techniques for classification and we have trained SVM with TF-IDF vectors.

TF-IDF + Random Forest: TF-IDF vector representation is used for words embedding with Random Forest classifier to train model for NER.

TF-IDF + SVMs: We used Support vector Machine (SVM baseline) that uses 2000d (TF-IDF) vector representation for our baseline model.

TF-IDF + AdaBoost: We used AdaBoost (AdaBoost baseline) that uses 2000d (TF-IDF) vector representation for our baseline model.

CountVectorizer + KNN: Aside from TF-IDF vectors, we did some experiments with CountVectorizer. We used sklearn's 30d CountVectorizer along with KNN.

CountVectorizer + Random Forests: We used sklearn's 1000d CountVectorizer along with Random Forests (RF) with rbf kernel and the C value of 100.

CountVectorizer + SVMs: We use sklearn's 1000d CountVectorizer along with Support Vector Machines (SVMs) with rbf kernel and the C value of 100.

CountVectorizer + AdaBoost: We use sklearn's 1000d CountVectorizer along with AdaBoost with rbf kernel and the C value of 100.

Word2Vec + KNN: In addition, we also use Word2Vec 200d embeddings trained on the dataset for 100 epochs with context window of 5, ignoring the tweets where minimum word count is less than 2, and the baseline model is KNN.

Word2Vec + Random Forests: We use Word2Vec 200d embeddings trained on the dataset for 100 epochs with context window of 5 with Random Forests.

Word2Vec + SVMs: We use Word2Vec 200d embeddings trained on the dataset for 100 epochs with context window of 5 with SVMs.

Word2Vec + AdaBoost: We use Word2Vec 200d embeddings trained on the dataset for 100 epochs with context window of 5 with AdaBoost.

Results of the baseline models are shown in table 4.5. KNN with TF-IDF gave the best results with the accuracy of 83%.

Classifier	Features	P	R	F1	Acc
KNN	TF-IDF	0.79	0.64	0.67	0.74
	CV	0.66	0.62	0.64	0.71
	W2V	0.83	0.68	0.71	0.76
SVM	TF-IDF	0.84	0.68	0.71	0.79
	CV	0.84	0.73	0.76	0.82
	W2V	0.79	0.72	0.74	0.79
RF	TF-IDF	0.85	0.73	0.76	0.83
	CV	0.83	0.73	0.76	0.82
	W2V	0.83	0.67	0.71	0.77
AdaBoost	TF-IDF	0.77	0.70	0.72	0.79
	CV	0.76	0.71	0.73	0.79
	W2V	0.69	0.65	0.67	0.72

Table 4.5: Baseline Classifiers and their Results

4.3.2 Proposed Methodology for NER

In this proposed approach we used different deep learning techniques to find the best possible solution for our problem. First, we used pre-trained FastText words embedding for Urdu with Bi-directional Gated Recurrent Unit (BiGRU) model for classification. Secondly, we used FastText

Urdu words embedding with a FastText like model for classification from the high-level Ktrain wrapper of Keras. Finally, we used Bidirectional Encoder Representations from Transformers (BERT) and fine tune them on our dataset before feeding it into a BERT model for classification.

4.3.3 Formal Solution Definition

Urdu is an agglutinative language. Each sentence is consists of multiple morphemes. Morpheme is a minimum semantic unit which has meaning but it can't be further divided. In this work we are going to tackle NER problem for Urdu language. The whole problem statement can be summarized in following points and figure.

Given a sentence S which contain some words as

$$\mathbf{S} = \{W_1, W_2, W_3, ...W_n\}$$

and entity types as

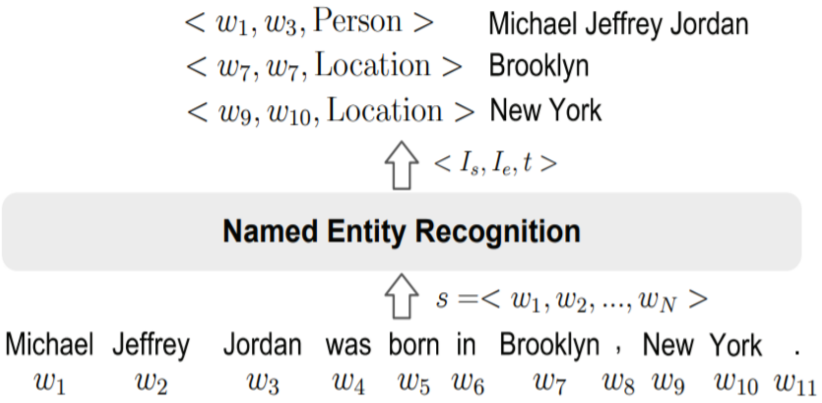
$$\mathbf{T} = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$$

there can be some words in a sentence which belongs to some specific entity **T**.

In this problem, the goal is to identify words **W** in a sentence **S** and classify each word W into it's correct entity type T as

$$\mathbf{W} = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$$

Above defined problem definition is illustrated in figure below.


< w1, w3, Person > Michael Jeffrey Jordan
< w7, w7, Location > Brooklyn
< w9, w10, Location > New York
< Is, Ie, t >
Named Entity Recognition
s =< w1, w2, ..., wN >
Michael Jeffrey Jordan was born in Brooklyn , New York .
w1 w2 w3 w4 w5 w6 w7 w8 w9 w10 w11

4.4 Urdu Named Entity Recognition Dataset

4.4.1 Dataset Collection

We collected dataset from BBC Urdu News and other Pakistani national news channels. We used **NewsAPI** which is an online platform to fetch news from multiple channels. We can get the news from *NewsAPI* according to categories. *NewsAPI* is an open source library and it is available for C/C++ and it returns the news in JSON format. ***Cython*** is a C++ wrapper for python which can be used for *NewsAPI*. We have collected and annotated the dataset of 48,650 words. Out of 48,650 words, 9,743 are named entities.

Named Entity	Train Set	Dev Set	Test Set
Person	2086	133	409
Location	1222	105	447
Number	1326	116	335
Organization	900	163	452
Designation	392	45	119
Time	209	22	100
Date	143	39	71
Total	**7178**	**623**	**1933**

Table 4.6: Final Dataset

4.4.2 Preprocessing

The collected dataset was in sentences and the dataset was noisy so we had to do the preprocessing to remove noisy materials from the data and also we had to tokenize a sentence into words because it would be easy for us to train a model on preprocessed and labeled words. Firstly, we removed the noise from a sentence. In noise removing, we removed the *html tags*, *punctuation*, *urls*, etc. After that, we tokenized the sentence into words and stored them in csv file. After performing above steps, our dataset was ready for next process which was manually labelling the data.

4.4.3 Feature Extraction

There are multiple techniques which can be used to extract features. There are techniques such as *statistical features*, *semantic features*, *lingustic features*, *word embedding*, etc. For **NER** different techniques have been used and the famous one is **word embedding**. In *word embedding* there are multiple techniques to get the word vector such as *FastText*, *GloVe*, *Word2Vec Embedding*, *TF-IDF Encoding*, *TF Encoding*, etc.

4.4.4 Classification

Most of the NER classification literature mainly focused on basic machine learning techniques like *SVM*, *KNN*, *Random Forest*, etc. To best of our knowledge, limited work has been done on NER using deep learning techniques. In this research work, we have experimented basic machine learning techniques such as *KNN* and *SVM*. After applying these techniques, we have shifted towards deep learning techniques so that we will be able to get distinguish between the ML and DL methods and it will be easy for us to know that which method will provide good accuracy.

4.5 Results and Discussion

In this section, we are going to present the results that we achieved on test set. First, we used FastText Urdu words embedding using maximum vector length of 25d and for classification we used FastText like model Joulin et al. [10] and we achieved 90.76% accuracy on the test data. Second, we used FastText Urdu words embedding using maximum vector length of 25d and for classification we used Bi-directional Gated Recurrent Units (Bi-GRU) and were able to get 83.3% accuracy. Third, we used BERT for both words embedding and classification and we achieved 95% accuracy on test set. The overall structure that we followed for NER is given in fig 4.2. We followed the same structure as it was presented in original BERT paper. The dataset was preprocessed with two techniques, we used *Ktrain tokenizer* and *BERT tokenizer* to get an idea that which tokenizer will work better with our problem. After tokenization, we converted the token into BERT input features consists of token-ids, segment mask and attention mask. We used the softmax cross-entropy loss function to measure the loss after each epoch and for hyper-parameter tuning we used BERT standard for hyper-parameter tuning.

Feature Extraction	Classification	Accuracy
FastText	FastText Like Model	90.76%
FastText	Bi-GRU	83.3%
BERT	BERT	**95%**

Table 4.7: Results of Various Embedding on NER Dataset

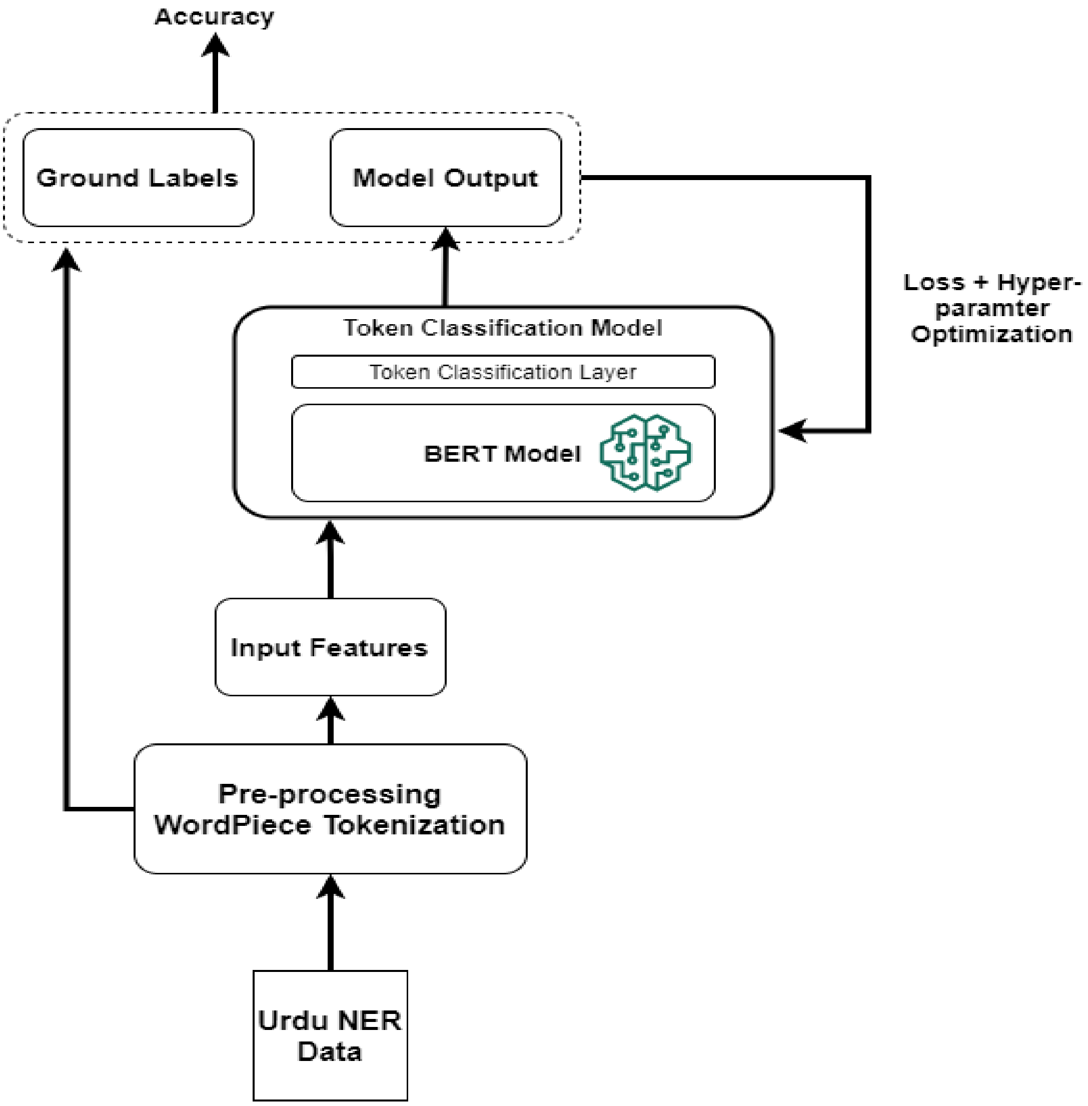


Figure 4.2: NER Solution Diagram

To get the best learning rate to train the BERT model for NER we ran a few epochs to find out the best learning rate and then plotted the results in such way that loss is on vertical axis and learning rate which is in log scale is on horizontal axis. As it can be seen from the plot that the minimum loss we get was at **2e-5** learning rate.

4.6 Evaluation Methodology

To evaluate the NER model, we split the data into train, validation and test set. We split the dataset in such a way that 70% of the dataset is being used in training, 20% for testing and 10% for validation as show in the pie chart 4.3.

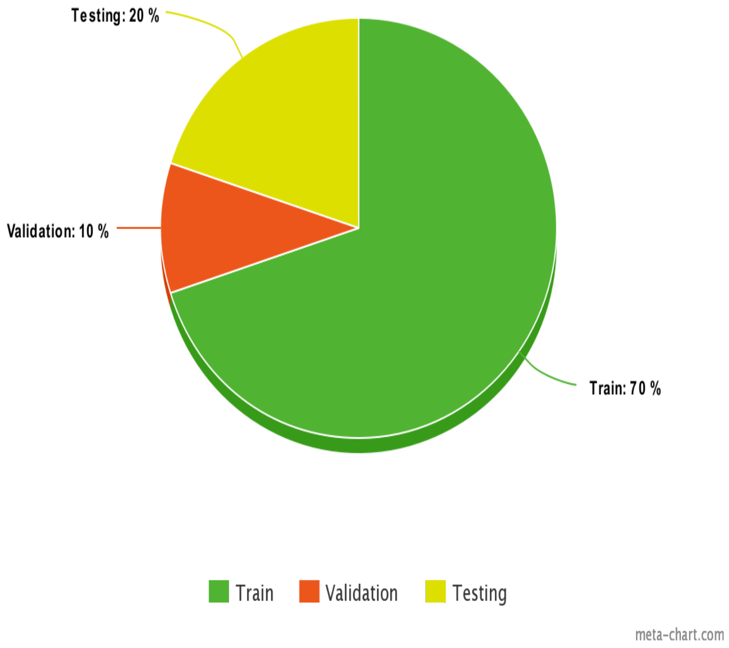


Figure 4.3: Dataset Distribution

4.7 Drawing Visuals

In this section, we will explain how we used our best models for *NER* and *Co-reference* to draw the objects on the screen. We will also give the brief details about the challenges that we faced and how we solved those problems. We used Unity game engine to process the data passing from the language input, the computer outputs a corresponding three-dimensional visual world where camera movements, physics, forces, collisions, animations, and motions are working together. We are proficient at intuition, language, imagination, and creativity, while the computer is proficient at computation, algorithm, logic, and memory, "Scribbling Speech using Urdu" created a mix of intuition and logic. It encourages creation with technology.

4.7.1 World Structure

It is a difficult task to Visualize human's "free speech" and because of the complexity, We have deconstructed the project into following states:

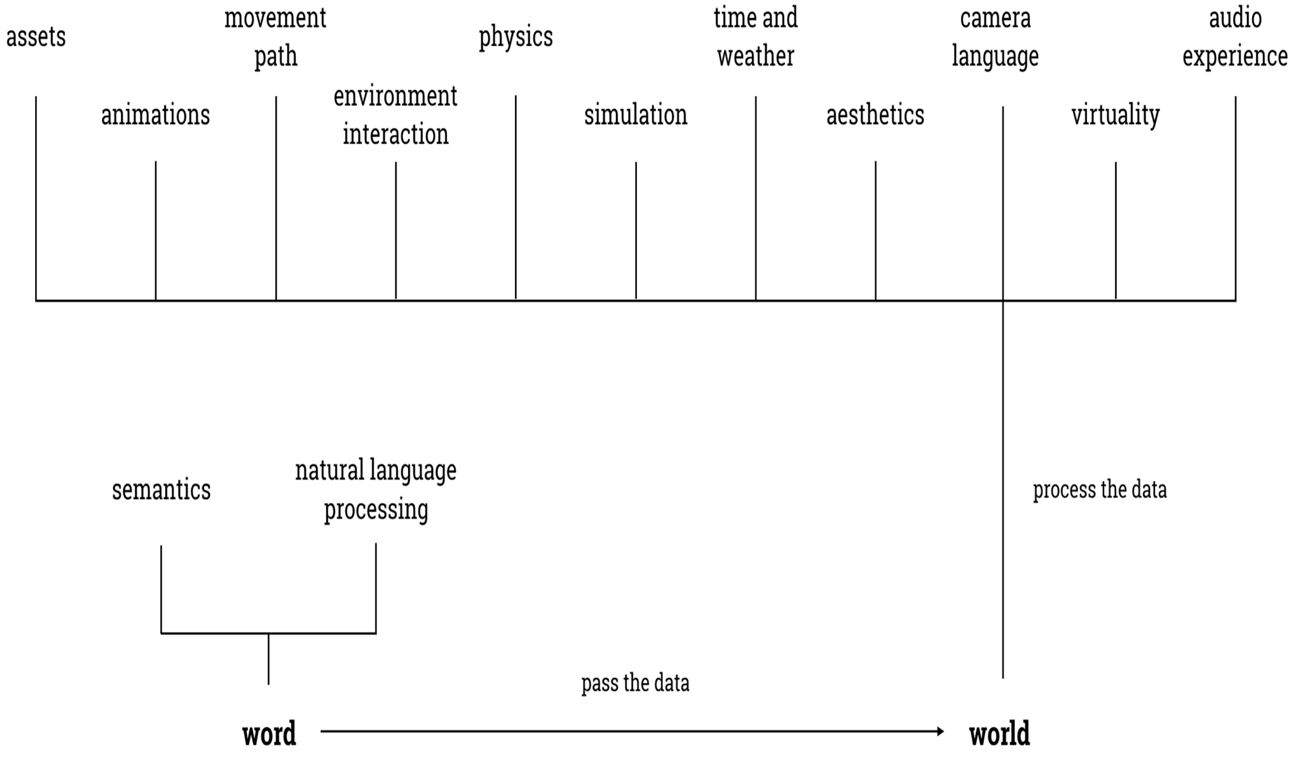


Figure 4.4: World Structure

4.7.2 Bringing Objects onto the Canvas

We call objects to be drawn as assets and assets is everything we have in a real 3D scene as natural environments, animals, people, objects, transportation and places, landmarks and so on. The computer will find the corresponding asset according to the detected ”nouns”.

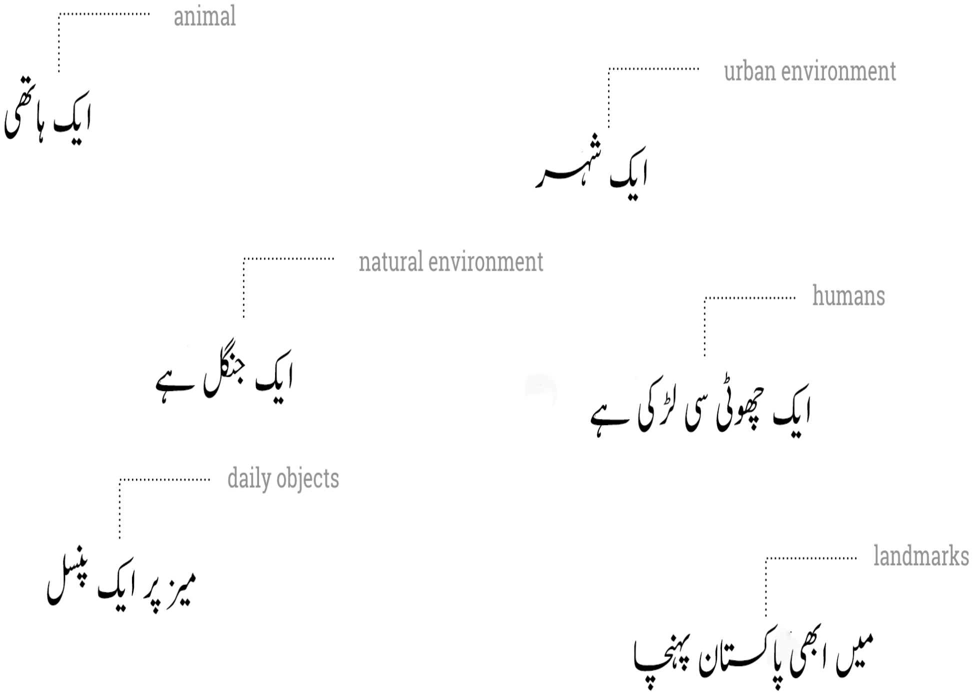


Figure 4.5: Word Assets

4.7.3 Animal Animations

In Unity3D, we use animators for adding animations to humanoid or generic rigged 3D models. We are using the Unity3D animator to run the animations according to detected *verbs*. In animator, start state is *idle* and every animation has a direct connection to every other connection so that any animation can be triggered from any point of animator.

4.7.4 Bringing Animations to the World

Animation word stems from the Latin word ”animātion” which means *bestowing of life*. To add animations in our 3D scene using natural language wasn’t an easy task but it can also increase the vividness of the scene.

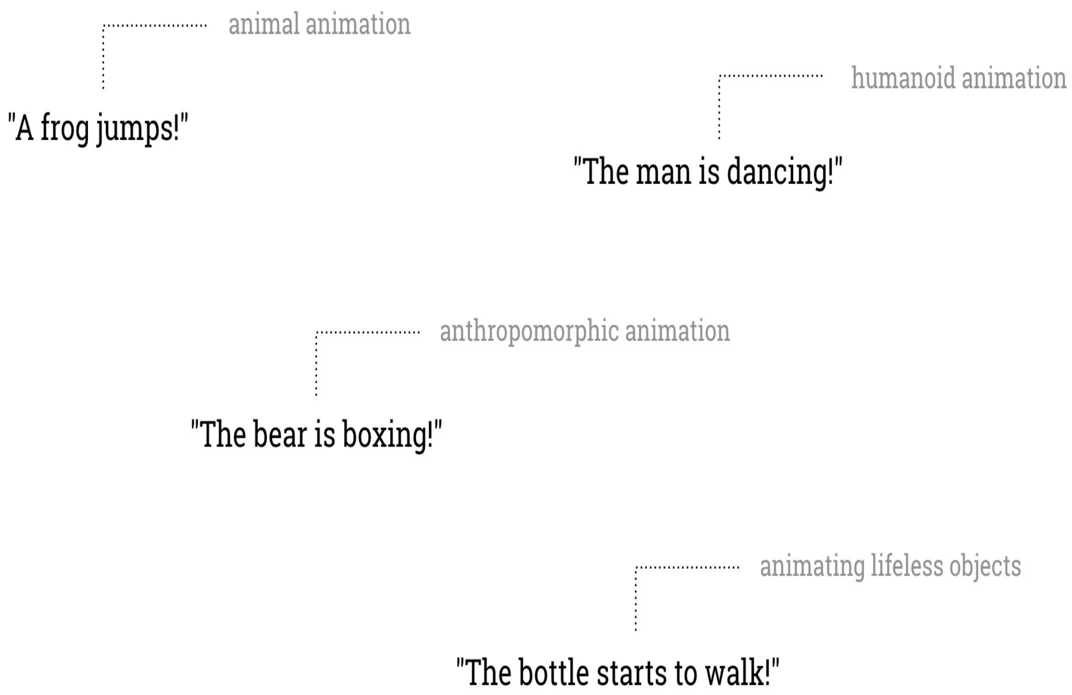


Figure 4.6: Animation Asset

4.7.5 Real-time Navigating

We are using the A* path finding to find out the walk-able area so that a particular object said by the speaker can walk. For example, if speaker is saying ”The girl walks in her room”, our AI system will calculate the walk-able area of the ”room” in real-time, this ensures that the girl won’t walk into tables and shelves.

4.7.6 Using Shadow to Visualize Time

In “Scribbling Speech Using Urdu”, we use the shadow to represent the sun and the time. At different times of the day, our shadow gets longer and shorter or may disappear. We can tell the time based on your shadow’s current length and angle. With this method, you are free to say ”Now it’s 7am in the morning.”

4.7.7 First Person Third Person

In “Scribbling Speech Using Urdu”, we use the shadow to represent the sun and the time. At different times of the day, our shadow gets longer and shorter or may disappear. We can tell the time based on your shadow’s current length and angle.

4.7.8 Adding a Virtual Layer

How do we visualize sentences like ٹی وِی پر خبریں سن رہا ہوں, ایک چھوٹا لرکا ٹی وی پر کارٹون دیکھ رھا ھے , “Scribbling Speech Using Urdu” will prepare tow virtual layers which often appear on “screens”, a child’s animation clip and a news report video clip.

4.7.9 Audio Experience

The main challenge that we faced was speech to text (STT) as we tried multiple STT systems for Urdu but none of them was accurate. In our final version, we have used, Google STT. There were some words which were not being accurately converted into text such as Google STT will convert word میز to میس sometime so we really spent most of the time to get false predictions so that if Google STT is giving a word میس then we are considering that word as a named entity and we have retrained our NER model for these miss-predicted words also.

4.7.10 Classifying Verbs to Assign Animation and Movement

We are using the distance between a group of words to find out which action to be taken according to user speech. Let's say if user said that *"the giraffe walk to the penguin"* then we will assign an animation and movement to that object which is *giraffe* in this case.

However, there are other verbs that can change the size of the object, verbs that can make the object talk, verbs that can make the object interact with the environment. So we have to classify the verbs according to their semantics.

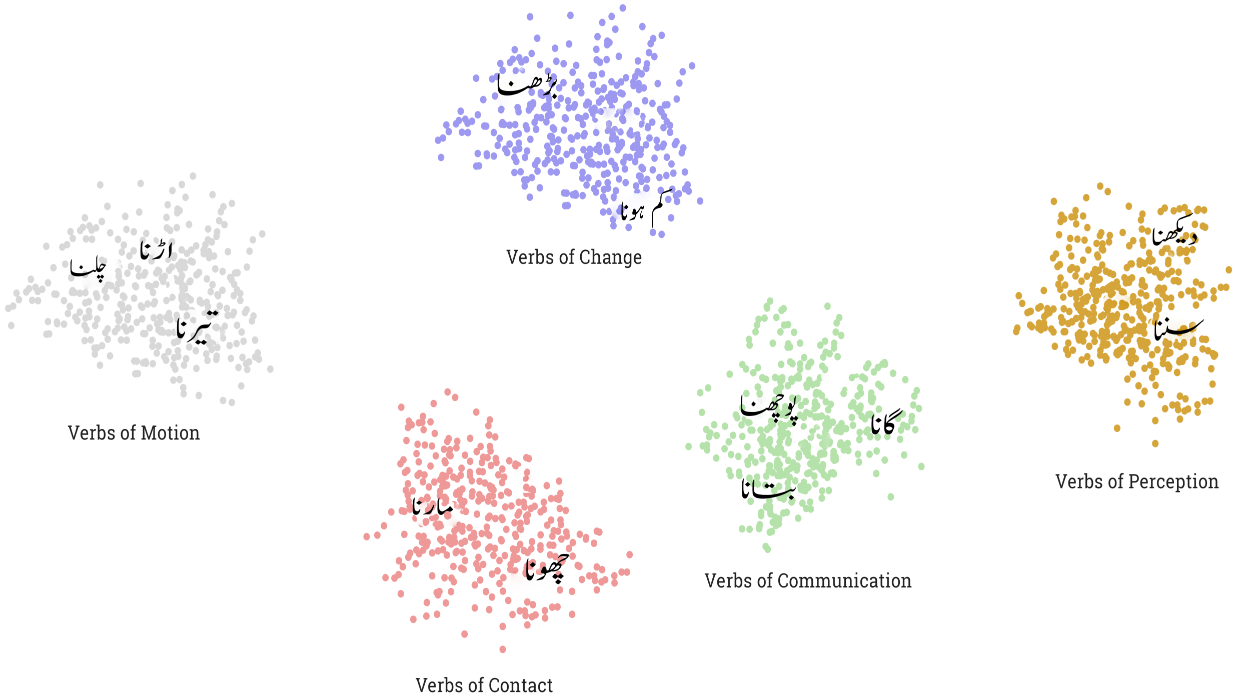


Figure 4.7: Animation Asset

Visual Results

After adding all the above modules, that We talked about, we were having an intelligent drawing system. Some of the visuals are shown below in the form of images.

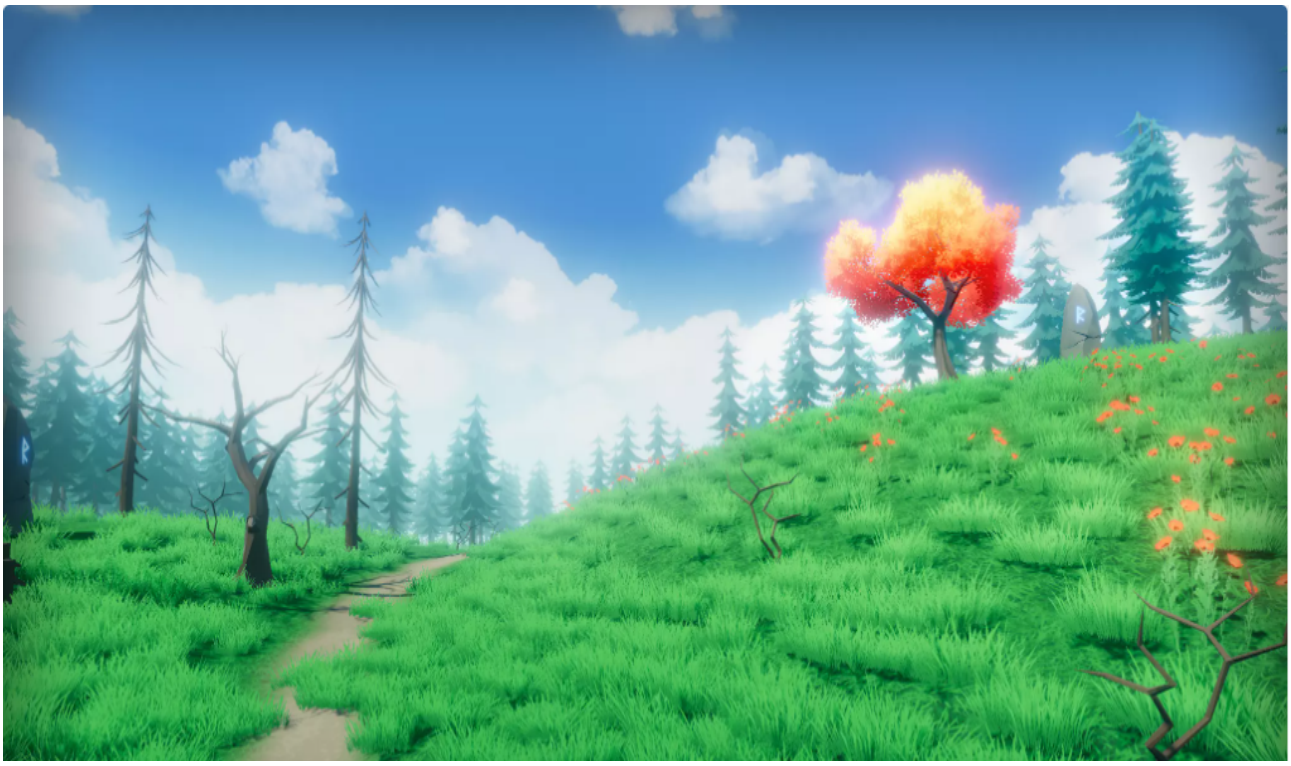


Figure 4.8: سامنے ایک درخت ہے۔

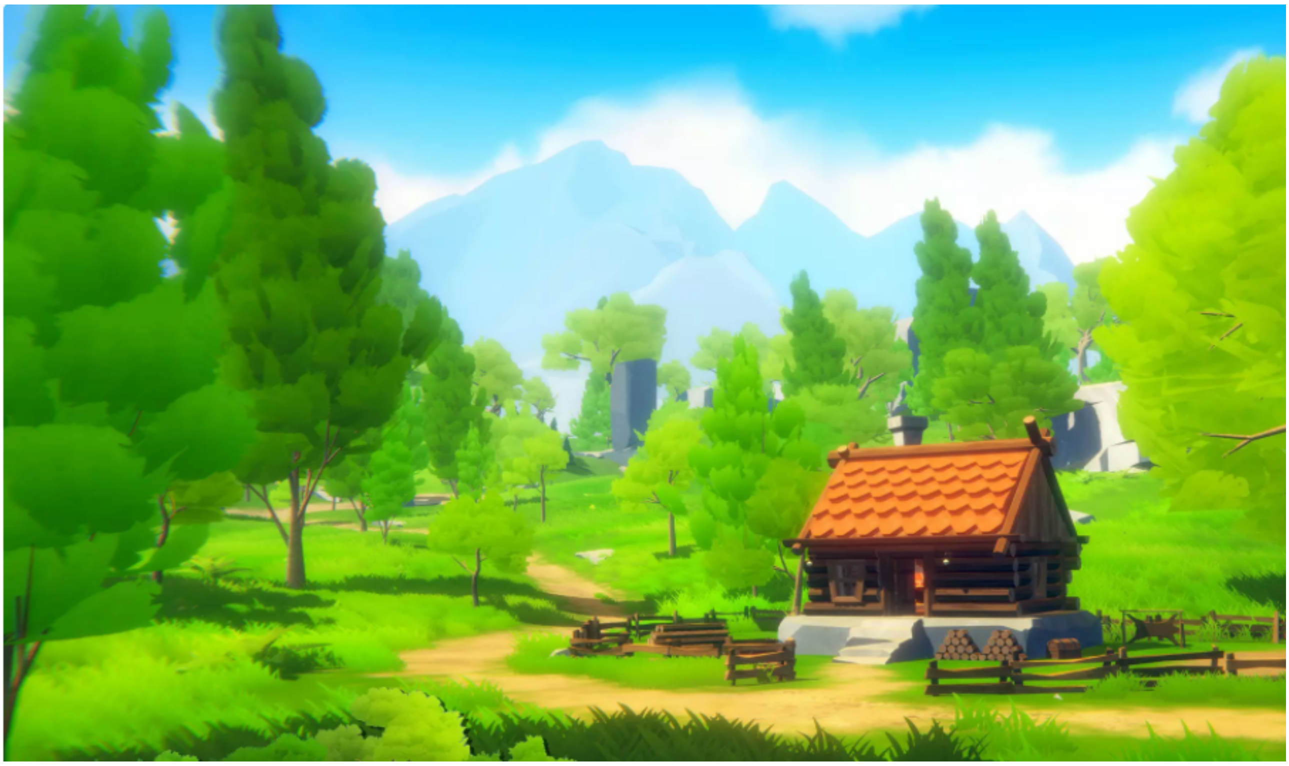


Figure 4.9: سامنے ایک گھر ہے۔

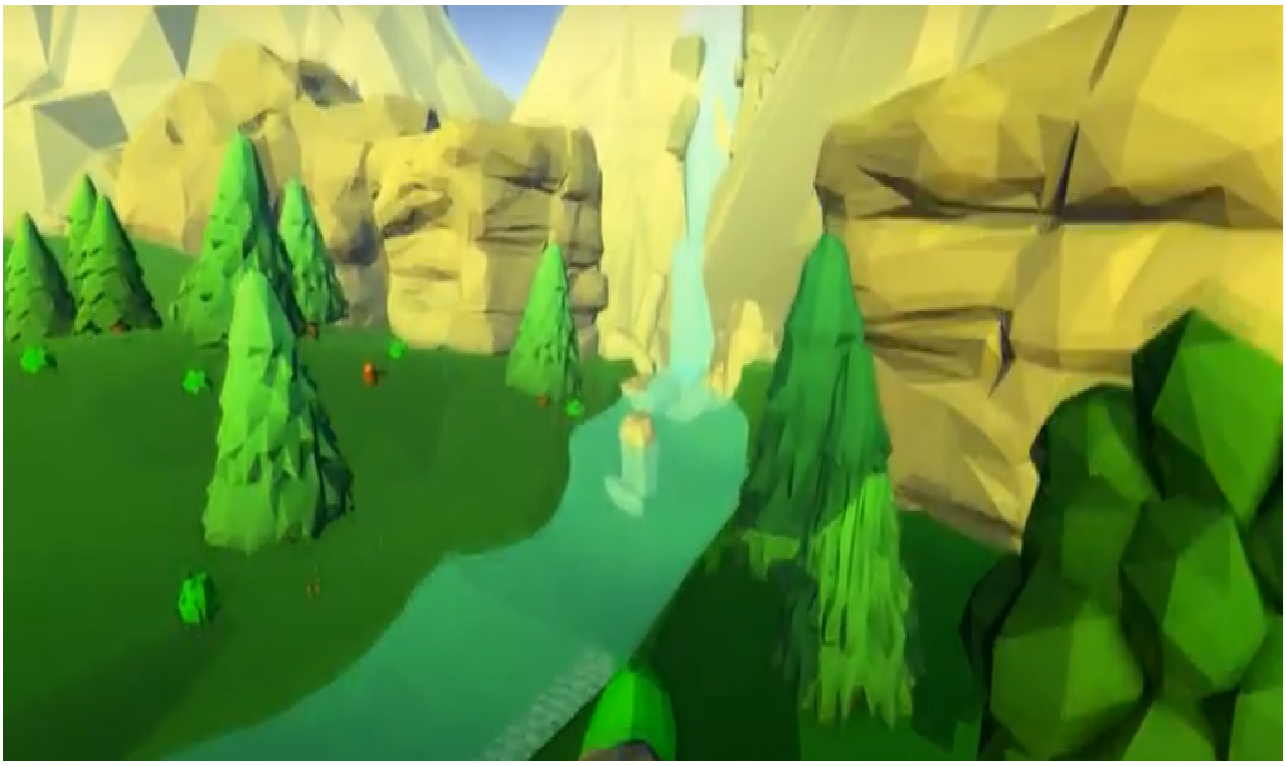


Figure 4.10: سمنظر میں ایک جھیل ہے۔

References

[1] David Nadeau, Peter D Turney, and Stan Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Conference of the Canadian society for computational studies of intelligence*, pages 266–277. Springer, 2006.

[2] Vincent Ng and Claire Cardie. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[3] Jerry R Hobbs. Resolving pronoun references. *Lingua*, 44(4):311–338, 1978.

[4] Wei Li and Andrew McCallum. Rapid development of hindi named entity recognition using conditional random fields and feature induction. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(3):290–294, 2003.

[5] Sam Wiseman, Alexander M Rush, and Stuart M Shieber. Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035*, 2016.

[6] Sobia Tariq Javed and Sarmad Hussain. Segmentation based urdu nastalique ocr. In *Iberoamerican Congress on Pattern Recognition*, pages 41–49. Springer, 2013.

[7] Kashif Riaz. Urdu is not hindi for information access. In *Workshop on Multilingual Information Access, SIGIR*, 2009.

[8] Lisa F Rau and Paul S Jacobs. Creating segmented databases from free text for text retrieval. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–346, 1991.

[9] Faryal Jahangir, Waqas Anwar, Usama Ijaz Bajwa, and Xuan Wang. N-gram and gazetteer list based named entity recognition for urdu: A scarce resourced language. In *Proceedings of the 10th Workshop on Asian Language Resources*, pages 95–104, 2012.

[10] Satoshi Sekine. Named entity: History and future, 2004.

[11] Andrew Borthwick and Ralph Grishman. *A maximum entropy approach to named entity recognition*. PhD thesis, Citeseer, 1999.

[12] Smruthi Mukund, Rohini Srihari, and Erik Peterson. An information-extraction system for urdu—a resource-poor language. *ACM Transactions on Asian Language Information Processing (TALIP)*, 9(4):1–43, 2010.

[13] Kashif Riaz. Rule-based named entity recognition in urdu. In *Proceedings of the 2010 named entities workshop*, pages 126–135, 2010.

[14] Niyu Ge, John Hale, and Eugene Charniak. A statistical approach to anaphora resolution. In *Sixth Workshop on Very Large Corpora*, 1998.

[15] Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139–162, 2020.

[16] Erik Cambria. Affective computing and sentiment analysis. *IEEE intelligent systems*, 31(2):102–107, 2016.

[17] Josef Steinberger, Massimo Poesio, Mijail A Kabadjov, and Karel Ježek. Two uses of anaphora resolution in summarization. *Information Processing & Management*, 43(6):1663–1680, 2007.

[18] Preusz Susanne, Birte Schmitz, Christa Hauenschild, and Carla Umbach. Anaphora resolution in machine translation. 1992.

[19] Luciano Castagnola. *Anaphora resolution for question answering*. PhD thesis, Massachusetts Institute of Technology, 2002.

[20] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*, 2017.

[21] Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, 2016.

[22] Yu Liu, Jianlong Fu, Tao Mei, and Chang Wen Chen. Storytelling of photo stream with bidirectional multi-thread recurrent neural network. *arXiv preprint arXiv:1606.00625*, 2016.

[23] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6329–6338, 2019.