UrduSpeech2Food: A Novel Speech Recognition System for Extracting Food Entities from Spoken Urdu Recipes

Students: Sana Fatima (sf06199), Shalin Amir Ali (sa06132),
Syeda Areesha Najam (sn05985)
Supervisors: Abdul Samad, Faisal Alvi

November 20, 2023

Abstract

This paper presents the development of a novel speech recognition system called UrduSpeech2Food, designed to extract food entities from spoken Urdu recipes. The system utilizes state-of-the-art speech recognition models and Name Entity recognition to convert spoken audio content into textual ingredient phrases, focusing on ingredient name, quantity, and unit. The research addresses the challenges of converting Pakistani Urdu recipes from speech to text. It also explores the potential applications of this technology in mobile and voice-controlled cooking appliances, highlighting its potential to improve accessibility for users. Overall, this report provides valuable insights into developing an efficient and accurate Urdu speech recognition system that can extract food entities from spoken Urdu recipes, with significant implications for the language processing and food technology field.

1 Introduction

Urdu language processing, particularly in the domain of food and nutrition, has been an area of limited research. This is primarily due to the lack of properly trained Named Entity Recognition (NER) models for Urdu, the limited availability of data in Urdu, and the absence of proper recipe listings in the language. In addition, the existing audio models for Urdu speech-to-text conversion are not very effective, further complicating the development of an Urdu speech recognition system that can accurately extract food entities from spoken Urdu. Our project aims to bridge this gap by developing an efficient and accurate Urdu speech recognition system to process audio, convert it to text, and extract food entities from spoken Urdu recipes.

The system has two primary components: audio-to-text conversion and named entity recognition (NER). Using pre-trained models, the audio-to-text conversion component will convert spoken Urdu audio into a text format. The NER module will then analyze the resulting text to identify and extract ingredient phrases, including the ingredient's name, quantity, and unit of measurement.

The purpose of this paper is to propose a system for converting Urdu speech to text, specifically for Pakistani food recipes. The system would be useful in transforming unstructured recipe data into structured information by identifying food entities, which would assist individuals in comprehending recipes more effectively.

2 Literature Review

The literature review for this project focuses on two main areas: Urdu speech recognition and ingredient identification in the recipe domain. The aim is to explore the current state of the art in these areas, highlighting various approaches and techniques used in both domains. In recent years, speech recognition systems have gained significant attention due to their potential to improve human-computer interaction. However, researchers face a significant challenge in this area due to the lack of resources for low-resource languages such as Urdu, which is spoken by millions of people but has not received enough attention in the field of speech recognition.

2.1 Urdu Speech-to-Text Recognition

In recent years, the development of speech recognition systems has gained significant attention due to their potential to improve human-computer interaction. However, one significant challenge researchers Face in this area is the lack of resources for low-resource languages. Urdu, a language spoken by millions of people, is one such language that has not received enough attention in the field of speech recognition.

We know that the traditional approaches to speech recognition rely on hand-crafted features and require large amounts of labeled data, which can be difficult and expensive to obtain. To overcome these limitations, two different models, Wav2Vec and Whisper, are described below which use the method of self-supervised learning and have proved to be efficient.

Recent studies have emerged which propose that self-supervised learning can overcome the limitations of large amounts of labeled datasets by leveraging unlabeled data to learn powerful representations. A state-of-the-art wav2vec2 model is introduced, which is based on a contrastive task defined over a quantization of latent representations [1]. The framework masks the speech input in the latent space and solves this contrastive task jointly with learning the quantized latent representations. The architecture of wav2vec2 consists of three main components: a feature encoder, a quantizer, and a transformer-based context network. The feature encoder is a convolutional neural network (CNN) that processes the raw audio waveform to produce a sequence of high-level features. The quantizer maps these features to discrete codes, which are then used to define the contrastive task. Finally, the context network uses a transformer-based architecture to model long-range dependencies in the input sequence and predict the masked features.

The training procedure for wav2vec2 involves two stages: pre-training and fine-tuning. The model is trained on unlabeled data during pre-training using the contrastive task defined over the quantized latent representations. This stage is designed to learn powerful speech representations from raw audio data alone. The model is fine-tuned on transcribed speech data using supervised learning in the fine-tuning stage. The paper also analyzes wav2vec2's performance on several benchmark datasets, including Librispeech, CommonVoice 9.0, and TIMIT. It shows that this approach achieves state-of-the-art results on these datasets while requiring less labeled data than previous methods. A notable aspect of wav2vec2 is its ability to achieve high accuracy with limited labeled data. It is demonstrated that using only 10 minutes of labeled training data, or 48 recordings of 12.5 seconds on average, achieves a word error rate (WER) of 4.8/8.2 on test-clean/other of Librispeech. Lastly, it was suggested that by switching to a seq2seq architecture and a word-piece vocabulary, the performance of the model can further be improved.

The paper "Robust Speech Recognition via Large-Scale Weak Supervision" [2] addresses the problem of robust speech recognition, which involves accurately transcribing spoken language in noisy and diverse environments. To solve this problem, the paper proposes a weakly supervised speech recognition approach involving training models to predict audio transcripts on the internet. It introduces the idea of the Whisper models, which are designed to match human behavior more closely than existing speech recognition systems. Whisper models are trained on a broad, diverse audio distribution and evaluated in a zero-shot setting. This means they are not fine-tuned on any specific task or dataset but tested on new data without additional training. This approach aims to see if the models can generalize well to new scenarios and match human performance. It uses a large-scale dataset of 680,000 multilingual and multi-task supervision hours to evaluate the approach. This dataset is used to train the Whisper models. To test the generalization capability of Whisper models, the models were evaluated in a zero-shot transfer setting without using any training data for each dataset. This means that they do not use any data from the target dataset during training and only evaluate the model's performance on this dataset after training.
The results show that Whisper models approach human accuracy and robustness in a zero-shot transfer setting without any fine-tuning. Specifically, the smallest zero-shot Whisper model with only 39 million parameters has a word error rate (WER) of 6.7% on LibriSpeech test-clean, which is roughly competitive with the best supervised LibriSpeech model when evaluated on other datasets. When compared to human performance, the best zero-shot Whisper models roughly match their accuracy and robustness. One limitation of this study is that the large-scale dataset used may not be representative of all possible speech recognition scenarios.

Hence, alongside the use of traditional speech recognition techniques such as Hidden Markov

and Gaussian Mixture models, various deep learning techniques have been proposed to achieve improved results with Urdu speech recognition. Thus, we aim to use similar models to perform Urdu speech recognition.

2.2 Ingredient Identification from Recipes

Once recipe data in Urdu has been successfully collected, the next step is to extract meaningful information from the text to structure our data into food entities such as ingredients, quantities, units, and more. In this section, we discuss some of the most promising techniques used in ingredient identification from recipes and highlight some of the challenges researchers face in this area.

To achieve this, A study was conducted to align instructional steps with speech signals in cooking videos using Hidden Markov Model (HMM) and computer vision technology [3]. This approach can aid in producing a large corpus of aligned recipe-video pairs capturing cooking actions and objects. Although the paper focuses on matching recipe text with cooking videos, the insights and approaches presented in the reviewed paper may be useful in informing the development of speech-to-text conversion systems in the recipe domain.

While we focus on obtaining ingredient information from recipe text, no such attempt has been made in Urdu. However, methods proposed in past research suggest that Named-Entity Recognition models are the best choice to apply when classifying keywords and phrases into predetermined categories. In this regard, a novel food ingredient named entity recognition model called RNE (Recurrent Network-based Ensemble Methods) has been proposed [4]. This model is an ensemble-learning framework utilizing recurrent neural networks such as RNN, GRU, and LSTM. These are trained independently and combined to provide better predictions in extracting food entities such as ingredient names, products, units, quantities, and quantities states from recipes. The experimental results show that RNE achieves a higher F1 score than individual models, indicating that it can efficiently extract information from text-based food recipes, which could be used to support various food-related information systems.

We need more detailed information for ingredients other than food names when extracting the information. Hence recognizing extensive food entities would be a crucial task [5]. The aim of the paper was to create a food dataset with a rich set of information on each ingredient in the recipe by identifying food entities such as food, quantity, unit, cooking process, physical quality or state, taste, color, the purpose of use and, part of the ingredient being used. The state-of-the-art baseline Named-Entity Recognition (NER) models, specifically BERT, CRF, and LUKE were used to extract food entities on a dataset of 700 recipes. Each ingredient in Around 13,000 values was extracted for all food entities altogether with a 0.95 F1-score obtained from the BERT model.

However, all the related work in this domain has been done on recipes written in English which is the major limitation we aim to overcome. Hence, to cater to an Urdu-speaking population of Pakistan, we aim to extract food entities from unstructured ingredient phrases in Urdu, which would be a novel contribution.

3 Evaluation Measures

Based on past research, we plan to use Word Error Rate (WER) and Character Error Rate (CER) to test the accuracy of our Urdu Speech Recognition System. We would also consider partial matches from ASR while the Precision, Recall and F1-score evaluates the performance of ingredient extraction from recipes.

4 Experimental Setup

4.1 Dataset Collection

In this project, we curated a dataset of Urdu recipes to design a real-time Urdu speech recognition system that accurately and efficiently extracts food entities like ingredient names from recipes. Since there is no existing proper dataset for Urdu Pakistani recipes, we collected and recorded the

data manually.

We collected altogether 100 recipes from two sources: Hum Masala [6] and kfoods [7]. We obtained recipe data in picture format for Hum Masala, so we had to manually transcribe the text into digital form. For kfoods, we initially attempted to scrape the Urdu content automatically but encountered Unicode errors that rendered the resulting dataset unusable. Thus, we manually transcribed the text from kfoods as well.

Once we had the text recipes, we recorded audio files of different individuals pronouncing the Urdu text. This allowed us to obtain recordings with varying pronunciations, which is crucial for training a robust speech recognition system and testing existing ones. All the recordings were converted into wav format to be processed by models for the ASR task. The audio files we have ranged in duration from 13 seconds, the smallest, to 53 seconds, the largest. On average, the duration of the audio files is 30 seconds.

Once the data was collected, we determined a suitable information storage format. To accomplish this, we identified the key fields required for our research purposes. The resulting dataset includes the following fields, which were deemed essential for our analysis:

- Recipe Name
- Additional Phrase
- Ingredient Name
- Ingredient Quantity
- Ingredient Unit
- Source
- Audio Path

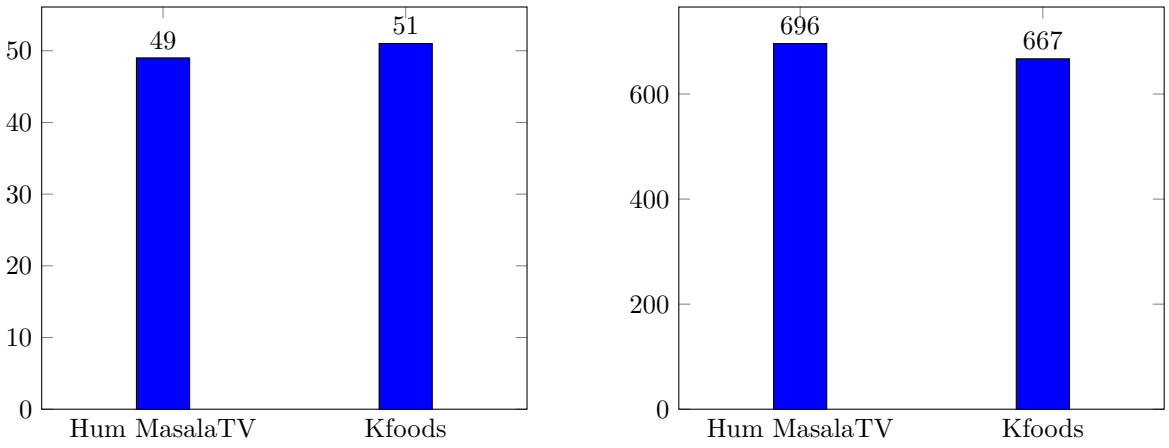
ID	Recipe Name	Additional Phrase	Ingredient Name	Ingredient Quantity	Ingredient Unit	Source	Audio Path
1	بلوچی گوشت	بلوچی گوشت بنانے کے اجزاء				kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		بون لیس مٹن	½	کلو	kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		ادرک لہسن کا پیسٹ	1	کھانے کا چمچہ	kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		ثابت لال مرچ	6	عدد	kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		ہری مرچ	3	عدد	kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		پسا ہوا زیرہ	1	چائے کا چمچہ	kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		ہرا دھنیا	1/4	گٹھی	kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		دہی	1	پاؤ	kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		نمک		حسب ذائقہ	kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		پسا گرم مصالحہ	1	چائے کا چمچہ	kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		پسا کھوپرا	2	کھانے کا چمچہ	kfoods	dataset\Balochi Gosht - kfoods.wav
1	بلوچی گوشت		تیل	3	کھانے کا چمچہ	kfoods	dataset\Balochi Gosht - kfoods.wav

Figure 1: A sample of the dataset collected for Urdu Recipes

The dataset shown in Figure 1 is stored in a structured format, facilitating easy retrieval and manipulation for further analysis. The resulting dataset includes text and audio files and is available for research upon request. The inclusion of audio files will enable researchers to evaluate speech

recognition models, specifically those designed for food entity extraction from Urdu text.

The dataset EDA involved analyzing two sources of ingredient phrases, Hum Masala and kfoods, that were curated for the task. There were 696 and 667 ingredient phrases from Hum Masala and kfoods, respectively. Additionally, the dataset included 49 recipes from Hum Masala and 51 from kfoods.



(a) Count of Recipes by Source

(b) Count of Ingredient Phrases by Source

Figure 2: Comparison of Counts by Source

Furthermore, the analysis identified 306 unique ingredients and 50 unique units in the overall dataset.

Entities	Count (unique)
Ingredient Name	306
Ingredient Unit	50

Table 1: Counts of Unique Ingredients and Units

4.2 Using Urdu Fine-tuned Urdu Speech-to-Text Recognition Models

To further prepare our dataset for model processing, the data was pre-processed using the following steps:

- Removed unnecessary characters such as commas, periods, quotation marks, colon
- Converted numeric values into their equivalent Urdu phrases
- Merged Additional Phrase, Ingredient Name, Ingredient Quantity, and Ingredient Unit into one string text associated with each recipe
- Normalized uneven spacing between words

Recipe Name	Audio Path	Text
بلوچی گوشت	dataset\Balochi Gosht - kfoods.wav	بلوچی گوشت بنانے کے اجزاء بون لیس مٹن آدھا کلو ادرک لہسن کا پیسٹ ایک کھانے کا چمچہ ثابت لال مرچ چھ عدد ہری مرچ تین عدد پسا ہوا زیرہ ایک چائے کا چمچہ ہرا دھنیا ایک چوتھائی گٹھی دہی ایک پاؤ نمک حسب ذائقہ پسا گرم مصالحہ ایک چائے کا چمچہ پسا کھوپرا دو کھانے کا چمچہ تیل تین کھانے کا چمچہ پیاز ایک عدد جائفل جاوتری پاؤڈر آدھا چائے کا چمچہ ثملہ مرچ ایک عدد کڑی پتے چند عدد پودین ایک چوتھائی گٹھی پسی لال مرچ ایک چائے کا چمچہ ہلدی ایک چائے کا چمچہ ابلے کٹے ٹماٹر دو عدد
فروٹی اسموتھی	dataset\Fruity Smoothie - kfoods.wav	فروٹی اسموتھی بنانے کے اجزاء آڑو ایک کپ درمیانی کیلے دو عدد دہی ایک کپ برف ایک کپ چینی ایک کھانے کا چمچہ
کرسپ اسپنچ ود سیلیڈ	dataset\Crisp Chicken with Salad - kfoods.wav	کرسپ اسپنچ و د سیلیڈ بنانے کے اجزاء چکن بریسٹ دو عدد کھیرا باریک سلائس ایک عدد ٹماٹر دو عدد باریک کٹی پالک ایک گٹھی مایونیز آدھا کپ آئس برگ آدھا پھول باریک چینی ایک کھانے کا چمچہ پسی کالی مرچ آدھا چائے کا چمچہ تیل فرائی کے لئے نمک حسب ذائقہ
کریم آلو	dataset\Cream Aloo - kfoods.wav	کریم آلو بنانے کے اجزاء آلو ایک کلو ہری پیاز دو عدد میدہ چار کھانے کا چمچہ وویسٹر سوس دو کھانے کا چمچہ فریش کریم چار کھانے کا چمچہ نمک حسب ذائقہ پسی کالی مرچ ایک چوتھائی چائے کا چمچہ پسی سفید مرچ ایک چوتھائی چائے کا چمچہ دودھ دو کپ
انڈے کی بریانی	dataset\Anday ki biryani.wav	انڈے کی بریانی بنانے کے اجزاء ابلے انڈے آٹھ عدد چاول دو پیالی باریک کٹی پیاز تین عدد دہی آدھا پیالی نمک حسب ذائقہ پسی لال مرچ ایک کھانے کا چمچہ ادرک لہسن کا پیسٹ ایک کھانے کا چمچہ پسا گرم مصالحہ ایک چائے کا چمچہ پودینہ آدھا گٹھی کچلی ہری مرچ چھ عدد لیموں چار عدد مکھن ایک کھانے کا چمچہ چھوٹی الائچی چار عدد تیل ایک پیالی چاول ابالنے کے لیے پودینہ آدھا گٹھی ہری مرچ تین عدد ثابت گرم مصالحہ تھوڑا سا کالا زیرہ ایک چائے کا چمچہ سفید سرکہ ایک کھانے کا چمچہ نمک حسب ذائقہ تازہ دودھ آدھا پیالی زردے کا رنگ ایک چٹکی

Figure 3: Dataset for Urdu Recipes.

Figure 3 shows the cleaned dataset where the complete recipe phrase and audio path columns are of our interest. To proceed with the task of Automatic Speech Recognition, we utilized state-of-the-art pre-trained models for Urdu Speech Recognition, specifically Whisper by OpenAI and Wav2Vec2 model by Facebook. To ensure ease of implementation, we utilized top fine-tuned versions of these models from the Hugging Face trained on the Urdu dataset and reported on the Papers With Code leaderboard for their respective datasets.

The anuragshas/wav2vec2-xls-r-300m-ur-cv9-with-lm [8] model is a fine-tuned version of Facebook/wav2vec2-xls-r-300m, trained on the CommonVoice 9.0 dataset by Mozilla Foundation. This fine-tuned model achieves a Word Error Rate (WER) of 31.72% and CER 10.50% with a loss of 0.4147. Similarly, for the Whisper model, we employed the ihanif/whisper-medium-urdu [9] model, a fine-tuned version of openai/whisper-medium trained on the CommonVoice 11.0 Urdu dataset also provided by Mozilla Foundation. This fine-tuned Whisper model achieves a WER of 26.98% with a loss of 0.4685. Although the latter model has a higher WER than the other, we will use both on our Urdu recipes dataset to compare the results.

To use them for inference, we used the **pipeline** abstraction function provided by Hugging Face, which encapsulates complex models into a simple API. We used a task-specific pipeline for automatic speech recognition to run both fine-tuned models. First, the audio files were converted into float arrays corresponding to the raw waveform of the speech signal as the models were trained on float audio arrays. This was achieved using **Audio** feature provided by **datasets** library in Python. Additionally, we used chunking in the pipeline by setting the **chunk_length_s = 30** which means it sends audio files greater than 30s into chunks of 30s, transcribes the chunks, and then combines them to produce one complete transcription. This was done the Whisper model is intrinsically designed to work on audio samples of up to 30s in duration.The chunking allows us to transcribe audio files with greater duration than 30s by sending large audio files into chunks of max 30s. However, using chunking can lead to poor inference of an audio file especially at the endpoints of each chunk. To prevent this, we used chunking along wih stride by setting an additional parameter i.e. **stride_length_s = (4,2)** where 4 is the left and 2 is the right stride length. As shown in Figure 4, Striding helps in doing inference on overlapping chunks so that the model actually has proper context in the center and a better transcription as a result. We also set batch size to 10. To increase the pipelines' computational power, we used the K-80 GPU provided by Google Colab as the models took more than 1 hour to run on the CPU for only 10 audio files. This shift helped us to process around 100 audio files in around 40 minutes.

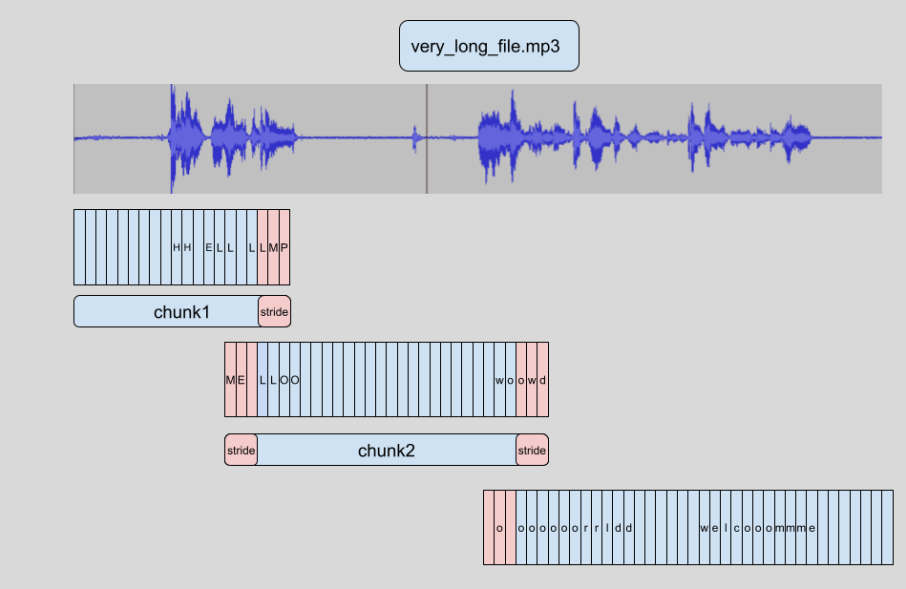


Figure 4: A visual representation of chunking with stride. Source: Adapted from [10]

4.3 Training a Named Entity Recognition (NER) Model for Urdu Ingredient Phrases

Prior to training a model, it is essential to perform data preprocessing to prepare the dataset for effective analysis. In our specific case, we conducted a series of preprocessing steps. Initially, we read the dataset shown in Figure 1, and subsequently, each row of the file was iterated over to extract the name of the ingredient, the quantity, and the unit, which was then merged to form a cohesive string.

We then performed cleaning on the string, which entailed removing any instances of the string "nan." Finally, the string was processed through the tag_input() function, which utilized regular expressions to extract and tag the quantity, unit, and ingredient in each string.

```
def tag_input(input_str):
    # Define the regex pattern
    pattern = r'(\S+)\s*(\d+(?:\.\d+)?)\s*(\S+)'

    # Extract the matches using the pattern
    matches = re.findall(pattern, input_str)

    # Create a list of tuples with the tagged entities
    for match in matches:
        ingredient = match[0]
        quantity = match[1]
        unit = match[2]
       entity = (input_str, {"entities": [(0, re.search('\d',
       input_str).start()-1, "Ingredient"),
        (re.search('\d', input_str).start(), re.search('\d',
        input_str).end(), "Quantity"),
        (re.search('\d', input_str).end()+1, len(input_str), "Unit")]})
        tagged_entities.append(entity)
    return tagged_entities
```

The regex pattern used in the function is defined as follows:

`(\S+)` matches one or more non-whitespace characters and captures them as the ingredient.
`\s*` matches zero or more whitespace characters.
`(\d+(?:.\d+)?)` matches one or more digits, optionally followed by a decimal point and one or more digits, and captures them as the quantity.
`\s*` matches zero or more whitespace characters.

(\S+) matches one or more non-whitespace characters and captures them as the unit.

If a single ingredient phrase is passed to the tag_input() function, it will index its ingredient name, quantity, and unit in the following format, as shown in Figure 5 and 6.

```
Input String:  ادرک لہسن کا پیسٹ 1 کھانے کا چمچہ
Sample output for one Ing Phrase:  ('ادرک لہسن کا پیسٹ 1 کھانے کا چمچہ', {'entities': [(0, 17, 'Ingredient'),
(18, 19, 'Quantity'), (20, 33, 'Unit')]})
```

Figure 5: Tagging the Ingredient Phrase using tag_input function

```
('تیل 4 کھانے کا چمچہ', {'entities': [(0, 3, 'Ingredient'), (4, 5, 'Quantity'), (6, 19, 'Unit')]})
('ہری مرچیں 2 عدد', {'entities': [(0, 9, 'Ingredient'), (10, 11, 'Quantity'), (12, 15, 'Unit')]})
('ٹماٹر 2 عدد', {'entities': [(0, 5, 'Ingredient'), (6, 7, 'Quantity'), (8, 11, 'Unit')]})
('ابلے نوڈلز 1 کپ', {'entities': [(0, 10, 'Ingredient'), (11, 12, 'Quantity'), (13, 15, 'Unit')]})
('میدہ 1 کھانے کا چمچہ', {'entities': [(0, 4, 'Ingredient'), (5, 6, 'Quantity'), (7, 20, 'Unit')]})
('وویسٹر سوس 2  کھانے کا چمچہ', {'entities': [(0, 10, 'Ingredient'), (11, 12, 'Quantity'), (13, 27, 'Unit')]})
('ہلدی 1  کھانے کا چمچہ', {'entities': [(0, 4, 'Ingredient'), (5, 6, 'Quantity'), (7, 21, 'Unit')]})
('تابت کالی مرچ 8 عدد', {'entities': [(0, 13, 'Ingredient'), (14, 15, 'Quantity'), (16, 19, 'Unit')]})
('کوکنگ چیز 2 پیکٹ', {'entities': [(0, 10, 'Ingredient'), (11, 12, 'Quantity'), (13, 17, 'Unit')]})
('تابت لال مرچ\xa0 6 عدد', {'entities': [(0, 13, 'Ingredient'), (14, 15, 'Quantity'), (16, 19, 'Unit')]})
```

Figure 6: Tagged Entities: Input to the NER

After the pre-processing, the dataset was randomly shuffled and split into train and test sets, with a ratio of 0.8 for the train set and 0.2 for the test set. Finally, the train set had 775 ingredient phrases, while the test set had 194 ingredient phrases.

Next, we created a model to extract ingredient names, quantities, and units from a given input phrase. For the said purpose, we chose SpaCy, an open-source Python library specifically designed for developing high-quality natural language processing (NLP) applications that are efficient and optimized for speed. It offers advanced features for text processing, including named entity recognition (NER), which involves identifying and extracting entities such as people, organizations, and locations from text. The NER capability in SpaCy is based on a statistical model that has been trained on a large corpus of annotated text data, using machine learning algorithms to learn patterns indicative of named entities.

To perform NER in SpaCy, it is necessary to load a pre-trained model that has been trained on the type of text data under consideration. However, in our specific case, we encountered issues when utilizing pre-trained models, as they were not effectively recognizing ingredients, quantity, and units as required. In contrast, these models performed well in extracting Parts of Speech (POS) from standard text. As a result, we decided to utilize a blank model, which allowed us to recognize and extract the relevant information.

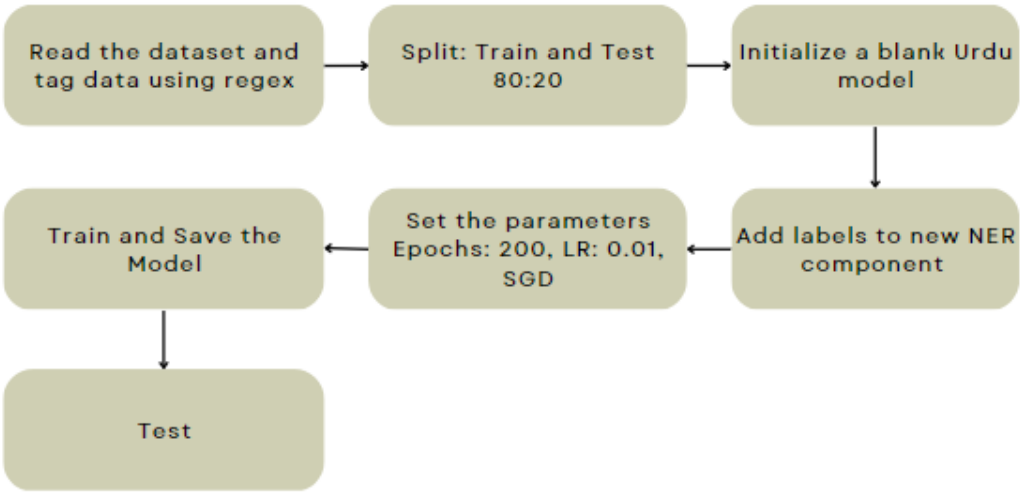


Figure 7: NER Model: Text to Food Entities

An Urdu natural language processing model was initialized and a new Named Entity Recognition (NER) component was created, consisting of labels such as "Quantity," "Unit," and "Ingredient." The NER model was trained for 200 epochs with a batch size of 4, utilizing a stochastic gradient descent (SGD) optimizer and a dropout rate of 0.35 to improve generalization. The training data was shuffled randomly and divided into batches to optimize the learning process.

The reason behind using the SGD optimizer is its common use for deep learning models, including neural networks used for natural language processing tasks like named entity recognition. It provides a good balance between accuracy and speed and can help the model to converge to a good set of weights for the task at hand.

4.4 UrduSpeech2Food: A combined model for Speech-to-text Recognition with NER

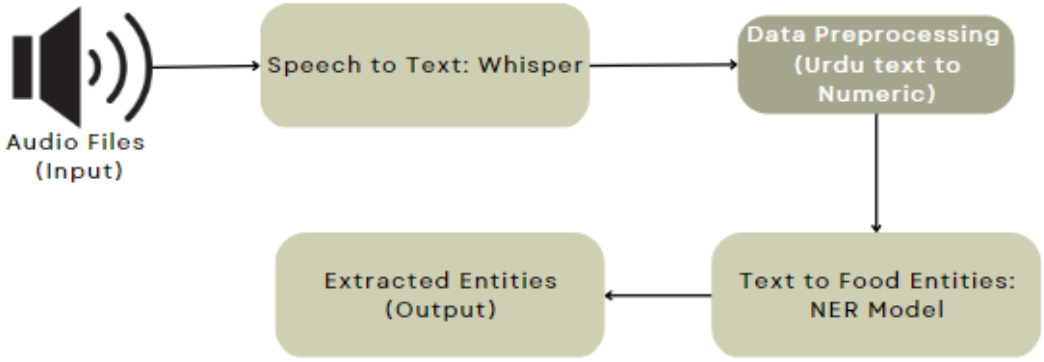


Figure 8: Combining the Models: UrduSpeech2Food

The combined process begins with the input audio files containing Urdu recipes. These audio files are passed through a pre-trained Whisper model, which effectively converts speech to text. Once the text is extracted, data preprocessing is performed to change the Urdu numbers in words to their corresponding numeric values. Next, the preprocessed recipe text is passed through a Named Entity Recognition (NER) model, which has been trained on the labels "Quantity," "Unit," and "Ingredient." Finally, the NER model outputs the food-related entities along with their respective labels.

5 Results and Discussion

5.1 Urdu Fine-tuned Urdu Speech-to-Text Recognition Models

We generated transcriptions for 100 recipes to compare the outputs obtained from the models, of which the first 5 are shown in Figure 9. Both models generated satisfactory results. However, there is a striking difference between the predictions obtained from both models when compared side by side.

Upon reading, one can identify that whisper-medium-urdu can detect even the slightest intervals between words and recognizes them as distinct words, while the wav2vec2-xls-r-300m-ur-cv9-with-lm mixes up the words into one single word. Another difference that can be seen is that the fine-tuned Whisper model generates text that contains commas to separate phrases within the sentence. Additionally, the whisper model generates comparatively similar output as the original text. To evaluate the results from both models, we used Word Error Rate (WER) and Character Error Rate (CER) as our metrics. Before that, we normalized outputs from both models by removing any punctuation or uneven spacing for a fair comparison. The results are shown in Table 2.

The whisper-medium-urdu model took more time to generate transcriptions for 100 audio files as compared to wav2vec2-xls-r-300m-ur-cv9-with-lm. According to CER, wav2vec2-xls-r-300m-ur-cv9-with-lm remained the best-performing model with a difference of 6.68% between both models. This shows that the fine-tuned wav2vec2 model captured Urdu characters with more precision. However, our area of interest is the WER of both models as it helps us gain an overview of the accuracy of recognition of complete words. Thus in terms of WER, whisper-medium-urdu undoubtedly gives us the best results with a WER of 56.08%. The WER obtained in implementing both models is higher than their reported WER.

Model	WER %	CER %	Processing Time (min)
whisper-medium-urdu	56.08	9.50	40.801
wav2vec2-xls-r-300m-ur-cv9-with-lm	60.64	25.8	1.986

Table 2: Performance comparison of the fine-tuned Urdu models

Recipe Name	Original Text	Whisper Predicted Text	Wav2Vec2 Predicted Text
بلوچی گوشت	بلوچی گوشت بنانے کے اجزاء بون لیس مٹن آدھا کلو اور ک لہسن کا پیسٹ ایک کھانے کا چمچہ ثابت الال مرچ چھ عدد ہری مرچ تین عدد پسا ہوا زیرہ ایک چائے کا چمچہ ہرا دھنیا ایک چوتھائی گٹھی دہی ایک پاؤ نمک حسب ذائقہ پسا گرم مصالحہ ایک چائے کا چمچہ پسا کھوپرا دو کھانے کا چمچہ تیل تین کھانے کا چمچہ پیاز ایک عدد جائفل جاوتری پاؤڈر آدھا چائے کا چمچہ شملہ مرچ ایک عدد کڑی پتے چند عدد پودین ایک چوتھائی گٹھی پسی الال مرچ ایک چائے کا چمچہ ابلی ایک چائے کا چمچہ ابلے کئے ٹماٹر دو عدد	بلوچی گوشت بنانے کے چمچے، تیل تین کھانے کے چمچے پیاز ایک اود، جائفل جوتری پاوڈر آدھ چائے کا چمچہ، شملا مرچ ایک اود، کڑی پتے چند اود، پودینا ایک چوتائی کئی، پسی لال مرچ ایک چائے کا چمچہ، ہلدی ایک چائے کا چمچہ، ابلی کئے ٹماٹر دو اود	بلوچی گوشت بنانے کے ساموں لیس مٹ دھکیل اورک لہسن کاپیسٹ ایک کھانے کا م ثابت لال مچ چھت ہری مر تین اوصا ہوا زیر ایک جائے مچا ہرا دنیا ایک جاتائی کردی دھی ایک با نمک ہسپانی کالیسہ گرم مثالی ایک جائے جاوسہ کو پرا دو کھانے کے جمے تیل تین کھانے کے جمے پیاز ایک عدد جائظیجوتری پاؤڈر اوا جائے گا چھشملہچرچ یعدد کڑی پتی چند دت پودینا ایک اتاری کرٹی پسی لال میٹ ایک جائے کا مچا لی ایک چھائے کا جما ہلی کئی ٹماٹر دو عدد
فرونی اسموتھی	فرونی اسموتھی بنانے کے اجزاء آڑو ایک کپ درمیانے کیلے دو عدد ونیلا ایسنس ایک چائے کا چمچہ	فرونی سمودی بنانے کے انزا اڑو ایک کپ ترمیانے گیلے دو اندہی ایک کپ برف ایک کپ چینی ایک کھانے کا چمچا	فرونی اس مودی بنانے کے غجزاڑو ایک کپ درمیانے کیلے دو عدد بھی ی ب برف ایک کب چینی ایک کھانے کا چمچاہ
کرسپ اسپنچ وو سیلیڈ	کرسپ اسپنچ وو سیلیڈ بنانے کے اجزاء چکن بریسٹ دو عدد کھیرا باریک سلائس ایک عدد ٹماٹر دو عدد باریک کٹی پالک ایک گٹھی مایونیز آدھا کپ آئس برگ آدھا پھول باریک چینی ایک کھانے کا چمچہ پسی کالی مرچ آدھا چائے کا چمچہ تیل فرائی کے لئے نمک حسب ذائقہ	کرسپ سپنچ ویٹ سالٹ بنانے کے اجزاء چکن بریسٹ دو اود خیرہ باری سلائز ایک اودٹماٹر دو اود باری کٹی پالک ایک کٹی مایونیز آدھا کپ ایزبرک آدھ پول باری چینی ایک چائے کا چمچہ پسی کھالی مرچ آدھا چائے کا چمچہ تیل تلنے کے لئے نمک ہزیڑائے کا	کرسپ سپنچ ویٹ سالٹ بنانے کے از چکن برس دو عدد کھیرا باری سلائس ایک عدد ٹماٹر دو عدد باری کٹی پالک ایک کٹی مایونیز آدھا کپ ایزبرک کب سزبرگ اوا پول باری چینی یک جائے کہ چچا پسی کالی مرچ اوا جائے ا مچا تیل لئے کیلئے نمک حسبزائے کا
کریم آلو	کریم آلو بنانے کے اجزاء آلو ایک کلو ہری پیاز دو عدد میدہ چار کھانے کا چمچہ وہیسٹر سوس دو کھانے کا چمچہ فریش کریم چار کھانے کا چمچہ نمک حسب ذائقہ پسی کالی مرچ ایک چوتھائی چائے کا چمچہ پسی سفید مرچ ایک چوتھائی چائے کا چمچہ دودھ دو کپ	کریم آلو بنانے کے عدزہ آلو آدھا کلو ہری پیاز دو عدد میدہ چار کھانے کے چمچے وہیسٹر ساس دو کھانے کے چمچے فریش کریم چار کھانے کے چمچے نمک حسب ذائقہ پسی کالی مرچ چوتھائی چائے کا چمچ پسی سفید مرچ چوتھائی چائے کا چمچ دودھ دو کپ	کریم آلو بنانے کے عصا آلو اوا ایکلوں ہری پیاز دو دردمی دا چار کھانے سے جمے وسٹر ساز دو کھانے کے جھر یشکریم چار کھانے کے جمے نمک ہسبزایکہ بسی کالی مرچ چوتھائی ای کا چمچ پسی سفید مچ چوتھائی جائے کا مچ ودو کپ

Figure 9: Comparison of original and predicted transcriptions from whisper-medium-urdu and wav2vec2-xls-r-300m-ur-cv9-with-lm

5.2 Named Entity Recognition (NER) Model for Urdu Ingredient Phrases

After training the NER model on the dataset as described earlier, it was tested on a test data to extract the "Quantity", "Unit", and "Ingredient" labels for each ingredient phrase.

The resulting outputs were then evaluated to measure the model's effectiveness using precision, recall, and F1 scores, which are commonly used metrics for evaluating the performance of a NER model. Following are the outputs for test data

```
Text: سونٹھ 2 ٹکڑے
Entities: [('سونٹھ', 'Ingredient'), ('2', 'Quantity'), ('ٹکڑے', 'Unit')]

Text: رائی دانے 2 چائے کا چمچہ
Entities: [('رائی دانے', 'Ingredient'), ('2', 'Quantity'), ('چائے کا چمچہ', 'Unit')]

Text: سادہ آٹا 1 پیالی
Entities: [('سادہ آٹا', 'Ingredient'), ('1', 'Quantity'), ('پیالی', 'Unit')]

Text: کٹی کالی مرچ 1 چائے کا چمچہ
Entities: [('کٹی کالی مرچ', 'Ingredient'), ('1', 'Quantity'), ('چائے کا چمچہ', 'Unit')]
```

Figure 10: Sample Ingredient Phrase on Trained Urdu NER

The precision of the model was calculated to be 0.86, indicating that 86% of the predicted entities were correct. Similarly, the recall was calculated to be 0.84, indicating that 84% of the entities were identified correctly. The F1 score, the harmonic mean of precision and recall, was calculated to be 0.85, indicating an overall satisfactory performance of the model. These results demonstrate that the NER model effectively identified the named entities for the ingredient phrases in the test dataset.

Evaluation Metrics	Scores
Precision	0.86
Recall	0.84
F1-score	0.85

Table 3: Evaluation for NER Model

5.3 UrduSpeech2Food: A combined model for Speech-to-text Recognition with NER

The evaluation of the UrduSpeech2Food module involved an assessment of the precision, recall, and F1-score. The results demonstrated that the module's performance was suboptimal, with a precision of 0.06, a recall of 0.05, and an F1 score of 0.05. The primary reason for this underperformance is the erroneous transcriptions from the Whisper model utilized for speech-to-text conversion.

Evaluation Metrics	Scores
Precision	0.06
Recall	0.05
F1-score	0.05

Table 4: Evaluation for UrduSpeech2Food Model

The Whisper model's Word Error Rate (WER) on our dataset was found to be 56.08%, which is a significantly high value. This indicates that the Whisper model struggles to accurately transcribe

the speech into text, leading to incorrect and incomplete data being passed to the NER model. This consequently results in low precision and recall values for the combined module.

The poor performance of the whisper-medium-urdu model on our dataset could be attributed to several factors, such as the talking speed, pause between each word, or the model may be trained on audio files with limited occurence of ingredients, quantities, or ingredient units as the model was trained on crowd-sourced data with no specific context. There are multiple strategies possible to enhance the performance of the whisper-medium-urdu model. These strategies include pre-processing the data recordings to enhance the audio quality of before feeding them into model, and fine-tuning the whsiper-medium-urdu model on the Urdu Recipe dataset mentioned in this paper. By employing these approaches, it is possible to achieve significant improvements in the overall performance of the whisper-medium-urdu model.

On the other hand, the NER model demonstrated satisfactory performance in identifying food entities and their respective labels. Thus, it is apparent that enhancing the accuracy of the Whisper model could substantially improve the performance of the combined module.

6 Future Works

The successful completion of this project will not only serve as a valuable resource for researchers and developers involved in Urdu language processing and has wide-ranging applications across various domains. This includes developing advanced natural language processing tools for Urdu, which can be applied in healthcare and the kitchen.

Moreover, we have planned to integrate this project with an ongoing application that aims to provide personalized meal planning services specifically for the Pakistani community and other communities with a taste for desi cuisine.

7 Conclusion

UrduSpeech2Food is a significant development in language processing and food technology. The system effectively extracts food entities from spoken Urdu recipes using advanced speech recognition and natural language processing techniques. Its potential applications in mobile and voice-controlled cooking appliances are vast, improving accessibility for users. This report demonstrates the feasibility of developing an efficient and accurate Urdu speech recognition system, paving the way for future research in this area.

References

[1] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision."

[3] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy, "What's cookin'? interpreting cooking videos using text, speech and vision," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 143–152. [Online]. Available: https://aclanthology.org/N15-1015

[4] K. S. Komariah and B.-K. Sin, "Enhancing food ingredient named-entity recognition with recurrent network-based ensemble (rne) model," *Applied Sciences*, vol. 12, no. 20, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/20/10310

[5] A. Wróblewska, A. Kaliska, M. Pawłowski, D. Wiśniewski, W. Sosnowski, and A. Ławrynowicz, "Tasteset–recipe dataset and food entities recognition benchmark," *arXiv preprint arXiv:2204.07775*, 2022.

[6] "Pakistan's no.1 food channel." [Online]. Available: https://www.masala.tv/

[7] "Kfoods recipes." [Online]. Available: https://kfoods.com/

[8] "Wav2vec2-xls-r-300m-ur-cv9-with-lm," accessed: 2023-04-10. [Online]. Available: https://huggingface.co/anuragshas/wav2vec2-xls-r-300m-ur-cv9-with-lm

[9] "Whisper medium urdu," accessed: 2023-04-10. [Online]. Available: https://huggingface.co/ihanif/whisper-medium-urdu#whisper-medium-urdu

[10] H. F. T. A. community building the future, "Making automatic speech recognition work on large files with wav2vec2 in transformers," Hugging Face Blog, 2022, [Online]. Available: https://huggingface.co/blog/asr-chunking.