CS-895: Big Data Analytics

Semester Project

Research Article

Title: Analysis of Textual Food Recipes with NLP

Authors:
Hassan Raza Bukhari
Muhammad Hasnain Naeem

Submitted to:
Dr. Muneer Ahmad

May 25, 2022

Abstract

Textual recipes are often incorrect because of missing ingredients, mistakes in referencing ingredients, and incorrect measurements such as weights, volumes, cooking times, and temperatures. This report explores textual recipes and how we can analyse these recipes to detect such errors using Artificial Intelligence and Natural Language Processing techniques. This project aimed to develop an automated solution that can parse, process, and analyze a recipe from the internet or a printed recipe to spot several distinct types of errors. The developed system can detect 5 distinct types of errors in the cooking recipes which are: (i) Ingredients being out of order. (ii) Missing Ingredients. (iii) Wrong Measurements. (iv) Incorrect Abbreviations. (v) Incorrect Cooking Times and Temperatures.The project flow includes data collection, pre-processing, ingredient parsing, named entity recognition, and recipe error detection steps. On testing the system, it is found that it can successfully detect recipe errors and achieved an average accuracy of 97%, precision of 0.96, recall of 0.96, and F1 score of 0.95. The major limitation of the system is that it is developed and tested on a small dataset. Future work may focus on improving preprocessing methodology, using a larger, high-quality dataset consisting of a large number of recipes, and using an ensemble of multiple models to improve the results. Also, the model can be deployed by wrapping it in a mobile or web app for a better user experience.

Keywords: Deep Learning, Natural Language Processing, Named Entity Recognition, Recipe Analysis

Acknowledgements

We would like to give thanks to our professor Dr. Muneer Ahmad for motivating us to work on a research article and supporting us throughout the journey.

Contents

Chapter 1

Introduction

Cooking recipes on the internet are becoming increasingly popular with more and more people trying online textual recipes for cooking ideas and guidelines. According to a survey, over 70% of adults are using the web (social media) for recipes instead of cookbooks [1]. Hundreds of people post their versions of a cooking recipe on various platforms, some have their blogs, while some others employ the use of social media to reach their audience. People use these recipes without initially considering the possibility of inaccuracies, inconsistencies, or incomplete information in these recipes. With this high reliance on the internet and social media for cooking guidelines, there is a question about the quality and correctness of these recipes.

This project explores the use of Natural Language Processing to automatically detect the common mistakes in the textual recipes so that they can be avoided. This project further demonstrates that even the most popular blogs can have these mistakes such that even if the recipes are followed to the letter, it might not result in the desired outcome.

1.1 Problem Statement and Motivation

People who follow the textual recipes either in cookbooks or on the web often complain about the results of the recipe not being as desired. The main reason for this is the mistakes made by authors when writing the textual recipes. Sometimes, there are typing mistakes such as mistyping abbreviations for measurement units, or the authors avoid mentioning things that they think are too obvious but for an inexperienced cook, those details are important. It is also common that some bloggers are just curating cooking recipes from different sources without having much knowledge about the recipe. Some of the most popular food blogs can have this problem as well, such as the Food & Wine editor-in-Chief admitting that she can't cook [2]. Correcting an erroneous food recipe is not necessarily just about correcting grammatical mistakes. There are several other mistakes that make the recipes erroneous. Some of the most common ones include incorrect cooking times and temperatures that result in undercooked or overcooked foods, ingredients missing from recipes that do not lead to the desired taste, ingredients not in order, that can confuse an inexperienced cook, incorrect measurements, and wrongly written abbreviations of measurements which can seriously affect the outcome of the recipe.

"it's more the rule than the exception that a recipe doesn't tell the whole story for the average cook" [3]

Detecting these mistakes in the natural text of the recipe instructions is challenging. The motivation of this project is to develop a system that can detect various problems in textual cooking recipes. This will benefit the authors and the readers both. The authors can use such a tool to make sure that the recipes do not have any errors that will make it hard for the readers to follow their recipes. While the readers can use this to check the recipes for any serious mistakes before starting to cook, thus saving themselves a good deal of time and effort in case of an erroneous recipe.

1.2 Aims and Objectives

This project aims to develop an automated solution that can analyze the textual recipes and detect some of the most common and important mistakes in the recipe instructions in a reasonable processing time with a minimum computational power requirement. The project has the following objectives:

- Researching for optimal techniques for recipe analysis
- Automated parsing of textual recipes and extraction of ingredients and other useful entities
- Designing and implementing algorithms to detect and interpret common errors in the cooking recipes
- Detecting several different types of common recipe mistakes
- Developing a command-line application that takes textual recipes as an input and reports the detected errors

1.3 Proposed Solution

The proposed solution will employ advanced Artificial Intelligence (AI) and text processing techniques including Natural Language Processing (NLP) to automatically identify and interpret problems in the natural text of recipe instructions. The solution will be based on the approach listed below:

- Researching blogs, forums, and social media to understand the common problems people are experiencing with the textual recipes and establishing a list of the most important ones that can lead to unintended outcomes in a recipe (based on experiences shared)
- Thoroughly researching the past literature on the subject matter, including literature on natural language processing for other similar unstructured or semi-structured datasets to get an understanding of possible techniques to be used and avenues that should be avoided
- Exploration of possible data sources and performing data collection activities using web scraping
- Developing the most important building blocks of the system which include accurate and robust recipe text parsing and Named Entity Recognition (NER) systems by exploring techniques like deep learning, text processing, etc
- Planning modules which will depend on the identified recipe problems to be detected. Using the building blocks to implement these modules
- Testing and Evaluating the system

  The success criteria for this system are the number of distinct recipe error types the application can identify, and also the accuracy with which it can identify the errors.

1.4 Structure of Report

The report structure is described in this section. The report consists of 7 main chapters. These chapters along with their brief description are listed below:

1. **Chapter 1 - Introduction**: This chapter describes the background, problem statement and motivation, aims and objectives, and the proposed solution of this project.
2. **Chapter 2 - Background**: This chapter explains the background of the project by first defining and explaining the important terminology and then discussing the works that are related to this project.

3. **Chapter 3 - Development**: This chapter goes into detail about the implementation of algorithms and modules for this project.
4. **Chapter 4 - Testing and Evaluation**: This chapter talks about the testing of the system on a test set and the evaluation metrics obtained.
5. **Chapter 5 - Tools and Technologies**: This chapter discusses the programming language and libraries used for the development of this project.
6. **Chapter 6 - Conclusion**: This chapter concludes the report by summarizing the work of the project, and discussing the problems faced and the limitations of the project.
7. **Chapter 7 - Future Work**: This chapter talks about the possible improvements in the system that can be done in the future.

Chapter 2

Background

The following chapter will start by defining and explaining some important terminologies necessary to understand the project. In addition to that, it will also include related work to this project and will discuss some interesting aspects of the research papers which are relevant to this project.

2.1 Natural Language Processing

Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) that is inspired by how humans can understand natural languages. The prime focus of NLP is to provide the machines with the ability to comprehend and interpret spoken and written language like humans.

"Natural language processing (NLP) is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language" [4]

NLP merges the concepts of computational linguistics (interpretation and generation of natural language using computers) with artificial intelligence methods such as statistics, deep learning, and machine learning.

NLP is being used in very common applications such as voice dictated web searches, GPS systems working with voice commands, automatic summarization of large texts, and also in the extraction of important pieces of information from a text, which is most relevant to this project. Examples of the extraction of information can be the interpretation of financial transactions to and from an organization from large financial documents which would take an extensive amount of time for a human.

NLP uses models similar to Machine Learning to perform all the different tasks which might include, but are not limited to:

1. Named Entity Recognition
2. Text Classification
3. Sentiment Analysis
4. Embeddings
5. Text Summarization

Only Named Entity Recognition (NER) is relevant to this project.

2.2 Named Entity Recognition

Named Entity Recognition (NER) is a process used in Natural Language Processing to identify and label various entities (pieces of information) within a text. For example, labeling the names of countries within a text and calling it entity type "country".

An entity can be a single word or can be made up of multiple words. For example, it can be "England" or "United States of America", both having a different number of words but they each represent a single entity which we can call a "country" as described in the previous example.

NER is often used to simultaneously categorize various different types of entities within the text. The most common ones are Geopolitical entities, persons, times, quantities, etc.

A use case of Named Entity Recognition is shown in Figure 2.1 below, it shows the identified entities in a simple cooking recipe.

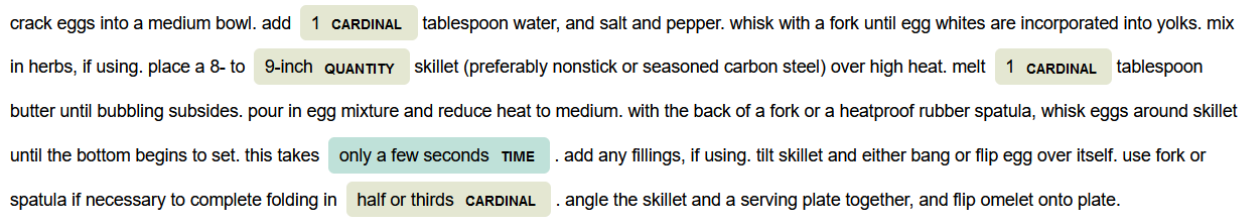

Figure 2.1: Named Entity Recognition Use Case

Named Entity Recognition can be done in two common ways:

1. Rule Based Named Entity Recognition
2. Deep Learning based Named Entity Recognition

The basic details of both are discussed below:

2.2.1 Rule based Named Entity Recognition

In Rule based NER, the entities are identified and categorized using patterns or rules. The textual patterns are defined in order to help in the identification of already known entities or to create a new entity that is previously not part of the model. The identification ability of the Natural Language Processing model is enhanced as more rules and patterns are fed to it.

2.2.2 Deep Learning based Named Entity Recognition

The deep learning based NER is very similar to the traditional deep learning tasks. It falls into supervised (in which labels are provided) machine learning task. The entities as well as their labels are supplied to the deep learning model to train it. The model uses these previously known entities and their corresponding labels to learn to classify new and unknown instances. Depending on the quantity and quality of data supplied to the model for training, it can detect and categorize the entities.

2.3 Related Work

Food recipe writers make some mistakes that can cause ambiguities and might not lead to the intended item discussed in the recipe. According to the Matching food and wine blog, the major cause for the mistakes in written recipes is that the chefs are not used to thinking in terms of recipes, when they are cooking their

thought process is different. Moreover, it might be needed to read the entire recipe carefully before starting to cook, to identify any problems in the recipe, while most readers do not do this. A very common problem can be mistyping the abbreviation of a quantity, such as putting "tbsp" (abbreviation for a tablespoon) instead of "tsp' (abbreviation for teaspoon) [5].

NLP has been used to interpret food recipes for various purposes in the past, one of them is recommendation of food recipes. Nilesh et al. used ingredient information within food recipes to recommend Indian cuisine recipes. It was based on the concept that there are several different Indian cuisine recipes that use the same or a very similar set of ingredients. This work uses the ingredients and already liked cuisine to recommend new cuisine recipes [6]. The relation of this research with our work is metadata and ingredient extraction from unstructured recipes. The data collection and preprocessing techniques used in this research served as an initial inspiration for structuring these parts of the project.

Processing a textual recipe is the most challenging part of this kind of research. Mori et al. used Machine Learning techniques to process textual recipes. This research aimed to convert the recipe into a workflow [7]. Some components of this research are reused in our project since there is a need to analyze the recipe in terms of a workflow, to be able to identify the order of ingredients within a textual recipe. Another important aspect covered in this research is the difference in general language and the recipe language which requires a need for adaptation for accurate results using annotated or partially annotated data. This research concludes that adaptation improves the accuracy of such a system in which analysis of textual recipes has to be done.

Hamon et al. used linguistic annotations and conditional random field (CRF) selection to extract ingredient names from food recipes. The research is performed on recipes written in the French language, and it is believed that the proposed methods are useful for the management and searching of recipes based on ingredients [8]. Some of the concepts used in this research can be reproduced to work on our project as well since ingredients need to be extracted to do the recipe analysis.

Hamon et al. used linguistic annotations and CRF selection to extract ingredient names from food recipes. The research is performed on recipes written in the French language, and it is relieved that the proposed methods are useful for the management and searching of recipes based on ingredients . Some of the concepts used in this research can be reproduced to work on our project as well since ingredients need to be extracted to do the recipe analysis.

There has been much work done in the named entity extraction for information related to food. A survey on this topic is written by Popovski et al. The survey goes into detail about four methods which are NCBO, FoodIE, SNOMED CT, and FoodOn. The experiments are performed using a manually annotated dataset consisting of 1000 recipes. The experimental results show that FoodIE outperforms other methods [9]. FoodIE is a method used to extract food information using rule based entity recognition. This method puts forward some interesting insights and methodology that can be used in our work. It is highly accurate, achieving 97% precision, 94% recall, and 96% F1 score on two separate test datasets.

To explore the avenue of unsupervised named entity recognition, Etzioni et al. did an experimental case study utilizing data from the web. Although the work is not done in the food domain, it is still insightful and can be reproduced for recipe datasets which are also available largely on the web [10].

Intelligent extraction of focus entities is not limited to recipe data, but work has been done in other domains as well. Some of these can serve as good starting points for our research. Similar work was done by Bodnari et al. as they used medical data to perform supervised named entity extraction. The system is able to obtain a 0.598 F-measure score under strict evaluation [11].

2.3.1 Research Gap

Several methods employ advanced Natural Language Processing and Machine Learning techniques to process and extract entity information, some are very specific to the food domain and are focused on food ingredients. This research work is focused on the intelligent extraction of entities including ingredients, most of which will be used for efficient searching and management of the policies. Utilizing some of the same principles of processing and extraction of entities, with minor improvements, our work proposes the use of extracted entities to analyze the recipe in terms of correctness. This system aims to identify the most common mistakes in written textual recipes using natural language processing techniques.

Chapter 3

Development

This section details the development of the recipe checker. The implementation is carried out in python because of the following reasons:

- Availability of several Natural Language Processing toolkits including NLTK and Spacy
- Availability of Web Scraping libraries that are used to gather data for the recipe checker
- Most popular language for scripting and text processing therefore has a huge user community

Further details for the choice of programming language and libraries is discussed in the Tools and Technologies chapter.

3.1 Project Flow

The figure below shows the overall project flow from data gathering to the detection of errors in the recipe. A brief description of each of the steps is discussed here, while the details explanation will be discussed separately:

1. **Data Collection**: Recipe and ingredient data is scraped from different sources, and open-source datasets are gathered to facilitate further steps of the development.
2. **Pre-Processing**: The data is cleaned and necessary conversions to required formats are done during this step.
3. **Ingredient Parsing**: Raw ingredient information is parsed into distinct and proper ingredient names to be used for error detection.
4. **Named Entity Recognition**: This step detects and interprets the various ingredients referenced in the recipe textual instructions using various techniques.
5. **Recipe Error Detection**: 5 different types of common recipe errors are detected and findings are displayed on a command line.

The project flow is visually described in the Figure 3.1

3.2 Data Collection

The details about collection of the datasets for this project are mentioned below:

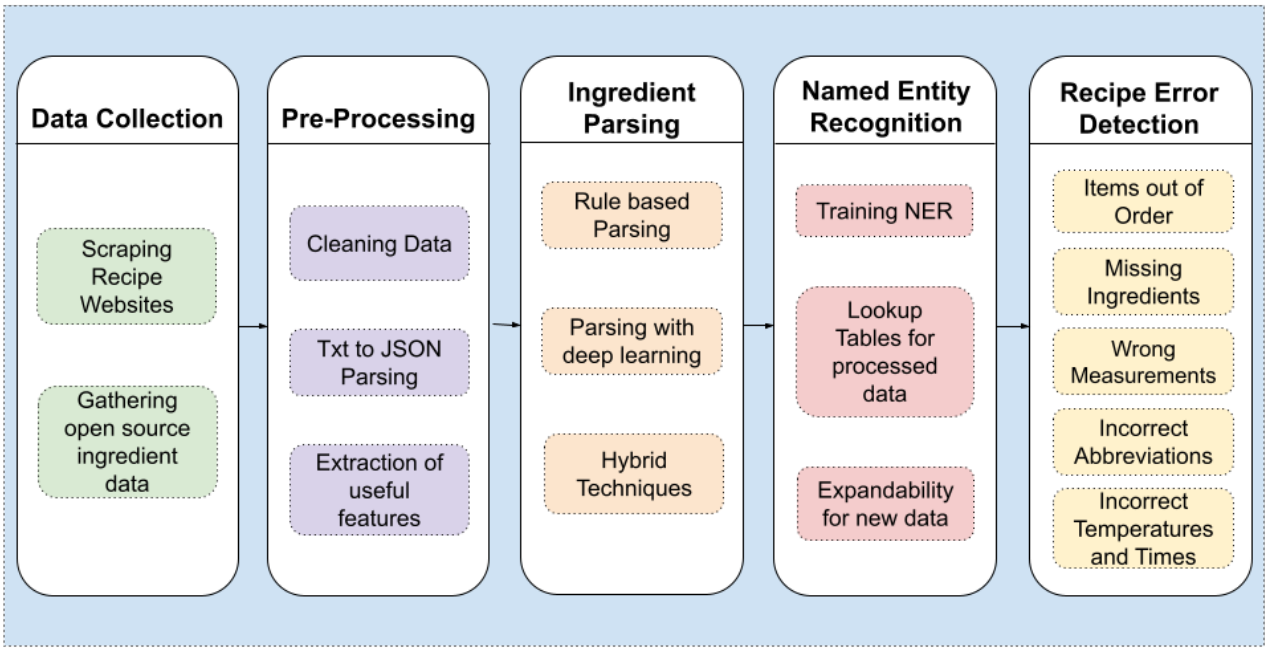

Figure 3.1: Project Flow

3.2.1 Scraped Datasets

There are two recipe instruction/ingredient datasets used throughout the project. The datasets are scraped from the web because of the unavailability of open-source recipe instruction datasets. Initially, the recipe data from onceuponachef.com [12] is scraped and after cleaning duplicates and problematic data items there are a total of 635 recipes. This dataset is very unstructured and requires extensive pre-processing to extract information from it. The data contains the textual instructions of the recipe, ingredient list, and metadata of the recipe including cooking times and meal servings. One major flaw in this data is that the author has not followed a standard approach in the ingredient list as well as the instructions. The ingredient list has a number of variations making it very impractical to extract ingredients from the unstructured ingredient list.

The second dataset is of much higher quality and it is scraped from NYTimes Cooking blog [13]. It has a total of 23 recipes. This data is also unstructured but follows a more standard approach for ingredient lists and thus facilitates the extraction of individual ingredients from the data. This dataset also contains recipe textual instructions, ingredient lists, and metadata.

3.2.2 Open Source Datasets

An open-source dataset from USDA Branded Food Products Database [14] is used to train an NER using deep learning to detect ingredient names. This dataset owes its existence to the partnership for public health: USDA branded food products database [15]. This dataset contains 1555131 instances of food items. There are several attributes in this dataset, although only the “description” is useful for NER training, which contains the names of various food ingredients.

In addition to this, another open-source data called the NYTimes ingredient phrase tagger public dataset [16] is used for training the Named Entity Recognition system using deep learning. There are 179207 instances in this dataset having the following relevant attributes:

- input (The unstructured input of ingredient name)
- name (The actual ingredient name that is to be extracted from the unstructured input)

- qty (Quantity of the ingredient)

There are several other attributes as well which are not relevant to this project.

3.2.3 Self Generated Datasets

Some datasets are self-generated as needed. The details of these are discussed below:

- The ingredient lists from scraped recipe datasets are used to parse ingredient names using multiple techniques each forming a new dataset.
- Firstly, a text processing based approach is used to parse ingredients from the ingredient lists which forms a dataset of ingredient names.
- Deep learning based approach is used to generate a dataset of ingredient names.

3.3 Pre-Processing

In pre-processing, the raw dataset is converted into a format that is usable by further steps including Ingredient Parsing and Named Entity Recognition. The scraped datasets have a number of problems including inconsistencies in the ingredient list format, problems with the format of instructions and variability of data features available from different sources. Most of these problems are addressed using automated python scripts. After the inconsistencies are handled, then the textual recipe is parsed and converted to JSON format. The reasons are doing this are:

- Parsing the textual recipe extracts the ingredient names, textual recipe, metadata.
- The recipe instructions are in steps, which are modified to have a full instruction component, which will be useful in some modules.
- Conversion to JSON gives a standard format that can be used efficiently in the next steps. Other formats can be used for this purpose as well. JSON is chosen because it is a standard format for semi-structured data. JSON is also very easily usable in python.

3.4 Ingredient Parsing

The ingredient lists with the recipe are raw and unstructured. It is easy for humans to read it and figure out the actual ingredient being referenced. Although, for a machine, there is much difficulty to interpret it and therefore some parsing techniques have to be applied.

For example, a typical ingredient is referenced like this:

“1 pound asparagus, trimmed and thinly sliced on a diagonal (about 1/4-inch thick)”

The actual ingredient is ‘asparagus’, but there is a lot of other information available as well. Although the ingredient name is the most important piece of information in this, there are other useful things as well, such as the quantity and units which are helpful as well. There is a need to extract all of the useful information by parsing it and storing separately for future use.

There are three main methods that are adopted to do this:

1. Rule Based Parsing
2. Parsing with Deep Learning
3. Hybrid Approach

The details of each one of them are discussed below:

3.4.1 Rule Based Parsing

In rule based parsing, text processing is applied based on rules or patterns to parse the ingredients. Although the data is unstructured, there are still some patterns that can be seen in the ingredients. These patterns can help in defining rules to parse the ingredients and extract useful information. Some of these rules are:

1. Splitting the ingredient item by comma (,) and keeping the first half of the split because in all the ingredient items the text after the comma has additional details only. In the previous example only "1 pound asparagus" will be kept.
2. In the extracted part of the string, it can be seen that in all ingredient items the first word in the string is always the quantity of the ingredient and the second one is the unit of that quantity.
3. The third word is the actual ingredient name.

Although it seems very intuitive for the ingredient example used, it is not always going to work. After several other variations and having tradeoffs in some ingredients, a rule based parsing technique is used to extract the useful information from the ingredients list. Another possible approach is to keep an exhaustive list of possible ingredient names and match the ingredient list with the items on this list. If any match is found, then an ingredient name can be extracted. One drawback of this method is that whenever more than one ingredient is referenced in a single ingredient item, it can give inaccurate results.

3.4.2 Parsing with Deep Learning

The NYTimes ingredient phrase tagger dataset is used to train a deep learning model to parse the actual ingredient name by taking the raw ingredient name as input. The format of the trained model is important and the new input to be parsed has to be supplied in the same kind of format. In case of a different format or incomplete input, there can be inaccurate parsing. Although it is accurate for items which are present in the training set, it does not generalize very well on the unseen dataset because the raw ingredient inputs have much variation and long length.

3.4.3 Hybrid Approach

A hybrid approach is used in which the ingredients are partially parsed using rule based parsing to obtain a substring that is a better representative of the ingredient after removing the additional details. This rule is also applied on the NYTimes ingredient phrase tagger dataset, which provides a modified version of the dataset. This new dataset is used to train a deep learning model for parsing the ingredients. The inputs to the model for inference must be parsed partially using the same rule as used in the training set. The results obtained using this hybrid technique are better than the individual approaches. Therefore, this approach is adopted as the final method for parsing the ingredients.

3.5 Named Entity Recognition

To find the various errors in the recipe instructions, it is important to extract useful pieces of information from it, which include the following:

1. Ingredient names
2. Ingredient quantities and units
3. Unit abbreviations
4. Cooking Times

5. Cooking Temperatures

This is the most important building block of the project and several approaches are used to get optimal accuracy for detecting the various entities. The details of each one of them are given below:

3.5.1 Training with Open Source Datasets

Python's spacy library is used to train a deep learning model using USDA's branded food dataset [14]. The USDA has a large number of instances to train a new entity that is called "food". The purpose is to detect ingredient names in the food recipes by using the food item names in the open source dataset. The model achieved an accuracy of 98% on a self generated food sentence dataset (generated using items in the training set). Although, it overfits severely and put every instance into a "food" entity, thus making it impractical.

3.5.2 Other Entities to reduce overfitting

Another open source dataset [17] containing a million news headlines is used to have numerous other entities while training the model so that overfitting can be reduced.

After the model is trained on a combination of this dataset and the food dataset, results are slightly improved but not considerably. Thus, this interaction also did not provide acceptable quality of food ingredient detection. The accuracy for food items, in this case, is 97%.

3.5.3 Custom Single Worded Ingredient List

The datasets used in previous iterations had multi-word food items. It is observed that the results could be improved by having a single-worded ingredient dataset. The idea is to keep only key ingredients to accurately identify the ingredient. For example, "cheese" is used instead of "cheddar cheese". This dataset is generated using text processing techniques.

After training and evaluating the model on this dataset, better results are obtained. The ingredient entity detection has improved and the model is also to detect most of the ingredients in the textual instructions with less overfitting. Nevertheless, there are still many ingredients missed and some are falsely classified as ingredients.

3.5.4 Text Processing Method

In this approach, in addition to using deep learning models, rule based parsing is done to detect the various ingredients in the instructions. The custom ingredient dataset is used which is generated using NER trained on the NYTimes ingredient phrase tagger dataset. Then the ingredient items present in this list are used to detect ingredients in the recipe instructions using string matching. This approach provided better results compared to previous techniques, although it is observed that accuracy can be improved by using a better recipe dataset.

3.5.5 Better Recipe Data and Text Processing Method

The NYTimes Cooking blog is used to scrape a better quality recipe dataset. After using this dataset with the hybrid text processing and NER method, it provided acceptable results. Data collection from this blog has one additional advantage which is its high compatibility with the NYTimes ingredient phrase tagger dataset. A lookup table is generated in this method which contains the inputs for ingredient phrases and the suitable ingredient name corresponding to that. This table is updated as new data is added to the system. The advantage of such a lookup table is much higher processing speeds in case of a repeated inference.

All of this information is hard to extract using traditional text processing techniques like rule-based parsing because the instructions have long paragraphs and a fair amount of additional information in it as well. Furthermore, the textual instructions are very unstructured. It is therefore important to use a technique like Named Entity Recognition employing deep learning to detect the various entities in the recipe instructions.

The NYTimes ingredient phrase tagger dataset is used to train the NER for detecting ingredient names. The results are improved by also including a custom ingredient name dataset. In addition to ingredient names, the NER is enabled to detect other entities by including the custom parsed dataset that includes all the other entities (quantities and units, abbreviations). Spacy's built-in models are good in detecting times and temperatures, therefore it does not need much training.

3.6 Recipe Error Detection

Using the functionality of the previous steps, modules are developed to detect the error in a textual recipe. The following 5 different kinds of errors are detected:

1. Ingredients out of Order
2. Missing Ingredients
3. Wrong Measurements
4. Incorrect Abbreviations
5. Incorrect Cooking Times and Temperatures

All of these modules use the building blocks described earlier. The details of each of these modules are discussed below:

3.6.1 Ingredients Out of Order

This module finds the order of all the ingredients in the recipe and reports if any of the ingredients are out of order. Firstly, the ingredient list is parsed to extract the ingredient names. The original order is determined from the ingredient list and this order is maintained to be compared later. After that, NER is used to find out the ingredients which are referenced in the instructions, and their order is found. The original order is compared with the order of ingredients in the instructions. If any of the ingredients are out of order, then an out-of-order ingredient error is generated.

3.6.2 Missing Ingredients

This module finds the missing ingredients in the recipe. The ingredients could be missing in two ways:

1. Ingredients are present in the ingredient list but its reference is missing from the recipe instructions.
2. The ingredient is referenced in the instructions but it is missing from the ingredient list.

To cater to both of these cases, this module has two sub-modules each to handle one of the two cases. The first sub-module checks whether there are any ingredients in the ingredient list that are not referenced in the instructions. The approach to accomplish this is to store the list of all ingredients in the ingredient list and then check whether all of them are referenced by processing over all the steps of the instructions. If any ingredient is found to be missing, it is stored in a list that can be used to display the ingredients which are missing. The second sub-module checks whether there are any ingredients mentioned in the instructions that are not listed in the ingredients. The approach is to first extract all the possible ingredients in the instructions using the lookup table and NER and then compare that with the actual ingredient list.

3.6.3 Wrong Measurements

This module finds the reference of all measurements for the ingredients and then checks whether the measurements mentioned in the ingredient list for each ingredient are the same as all the different measurements referenced for that ingredient throughout the instructions.

A cooking specific measurement unit list was created by studying the data and the web for possible measurement units, this list is helpful in various parts of the module. It was found that not all the ingredients are measurable, therefore only those ingredients will be considered which are measurable in numerical values so that appropriate comparisons can be made. A function called 'find_measurable_units' was developed to accomplish this task. The functionality of the missing ingredients module is also utilized in this module to further refine the search to only those ingredients which are available in both the ingredient list and the instructions. Based on all this, a new component of data was generated called 'ing_units' which contains the measurement details of all measurable ingredients in a recipe.

After researching on the web, two lookup tables are created, one is a 'conversion table' which is used to convert between cooking units. The conversion table assumes the 'teaspoon' to be relative unity for all measurements, and then stores conversions based on this assumption. The other one is the 'value table' which is used to get numerical equivalent to special character representation such as $\frac{1}{4}$. A function called 'extract_measurements' was developed to extract all measurements associated with an ingredient in the instructions of the recipe. After extracting the measurements, they need to be parsed and for this purpose a measurement parser algorithm is used to get numerical values for both ingredient measurements in ingredient list and also in the instructions. Employing all of the sub modules mentioned earlier, it can be found whether there are any problems in the measurements of a recipe.

3.6.4 Incorrect Abbreviations

This module finds if there are any problems in the use of abbreviations for measurement units in the recipe instructions. The core idea is to extract the actual measurement units of each ingredient from the ingredient list, then traversing the entire recipe instructions to see whether all the references of abbreviations for the measurement of that ingredient are correct.

Some functionality is reused from previous modules. A lookup table is created for possible correct abbreviations for the measurement units being used.

3.6.5 Incorrect Cooking Times and Temperatures

This module finds whether there is any mistake in the cooking times and temperatures referenced in the recipe instructions. For most items, there is a defined range of acceptable temperatures and times. This information is used to make sure that all cooking times and temperatures for common ingredients fall within this acceptable range. This module is implemented with one assumption that the temperatures and times are analyzed for items being cooked instead of individual ingredients because it is impossible to track individual ingredients as they are not referenced individually in the instructions when temperatures or times are discussed in the instructions.

A lookup table is created for acceptable ranges of temperatures and times for common items by researching on the internet. The table was stored as a CSV and then later processed into a python object so that it can be used for the analysis later on. All the temperatures and times are extracted from the recipe instructions using NER. After extraction of the temperatures and times, they are compared with the lookup table of acceptable times and temperatures to identify any problems.

Chapter 4

Testing and Evaluation

The testing is done first for the evaluation of each module individually and then for the entire system. 23 cooking recipes from the NYTimes Cooking blog are used for testing. Sometimes the mistakes are manually inserted to evaluate the system in such cases when no mistakes are originally available for a particular module.

The evaluation will be done on the following criteria:

- How many recipes had a particular mistake and how many of them are correctly identified?
- How many of the recipes did not have a mistake and are incorrectly identified as a recipe with mistakes.

Furthermore, traditional Machine Learning metrics like accuracy, precision, recall, F1 and Confusion matrix will also be used to evaluate the system.

The metrics used for the evaluation of various modules are explained below:

- Accuracy: percentage of correct predictions .

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

- Precision: proportion of positive identifications that are actually correct.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

- Recall: proportion of actual positives that are identified correctly.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

- F1 Score: the harmonic mean of precision and recall, higher the better.

$$F1Score = 0\frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.4}$$

In above equations, TP indicates the number of true positives, TN indicates the number of true negatives, FP indicates the number of false positives, and FN indicates the number of false negatives.

The details about testing for each module are discussed below:

4.0.1 Ingredients out of Order

It is a very common problem with online textual recipes when ingredients are referenced in a particular order in the ingredient list, but in the textual instructions, they are referenced out of order, thus causing confusion for someone who is following the recipe.

A test set of 23 recipes is used, in which 3 of the recipes have ingredients out of order. Since this problem is common, there is no need to introduce problems. Even a popular cooking blog like NYTimes cooking blog also has 3 out of 23 recipes with out of order ingredients problems.

On running the system on this test set, the following results are obtained as shown in Table 4.1:

Table 4.1: Evaluation Results for Ingredients Out of Order Module

Number of Recipes with Mistake	**Number of Correctly Identified Mistakes**	**Number of Recipes with no Mistakes**	**Number of Incorrect Identifications**
3	3	20	0

The confusion matrix for this is shown below in Figure 4.1:

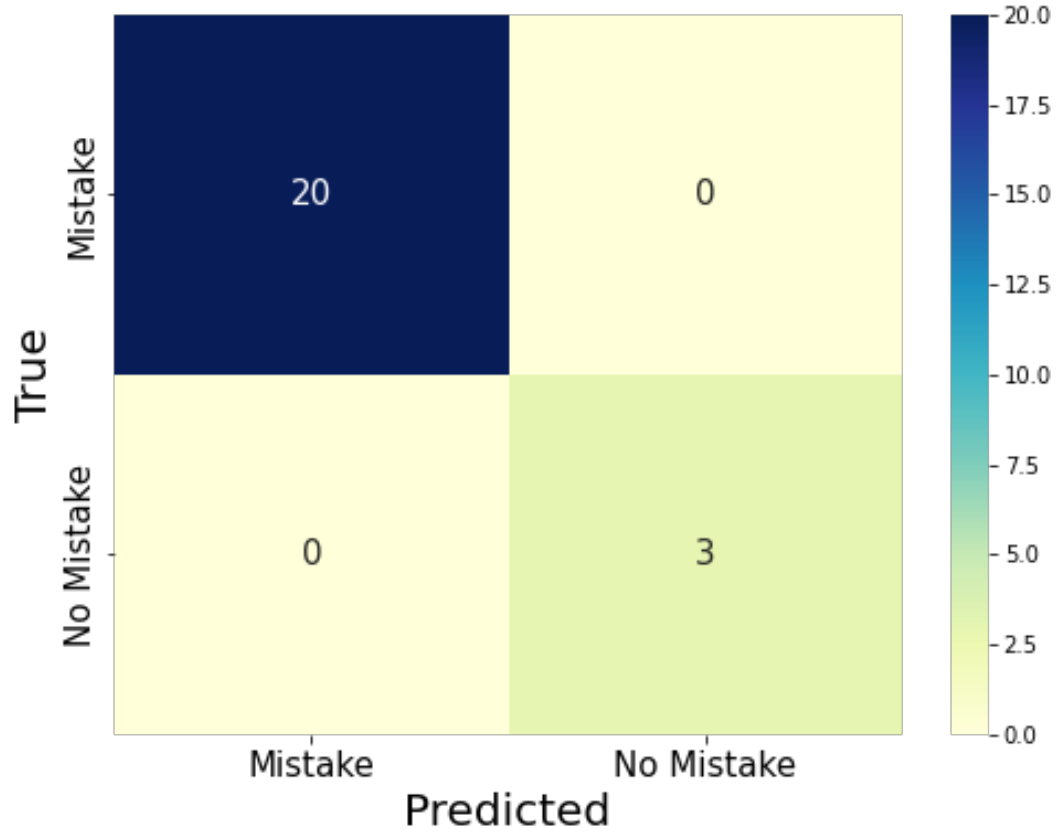

Figure 4.1: Confusion Matrix for Ingredients Out of Order Module

4.0.2 Missing Ingredients

If ingredients are mentioned in the ingredient list, but missing from the recipe instructions, it can be problematic for the person following the recipe and can lead to a wrong result. Furthermore, most people make purchases of the ingredients by looking at the ingredient list, and later on, if they find some ingredients which are being referenced in the instructions which are never listed in the ingredient list, then it can also lead to wrong results. The latter is much more common than the first.

In the test set of 23 recipes, 18 of the recipes had this problem, which shows that this is a very common problem even among famous blogs. The evaluation is shown below in Table 4.2:

Table 4.2: Evaluation Results for Missing Ingredients Module

Number of Recipes with Mistake	**Number of Correctly Identified Mistakes**	**Number of Recipes with no Mistakes**	**Number of Incorrect Identifications**
18	18	5	1

The confusion matrix for this is shown below in Figure 4.2:

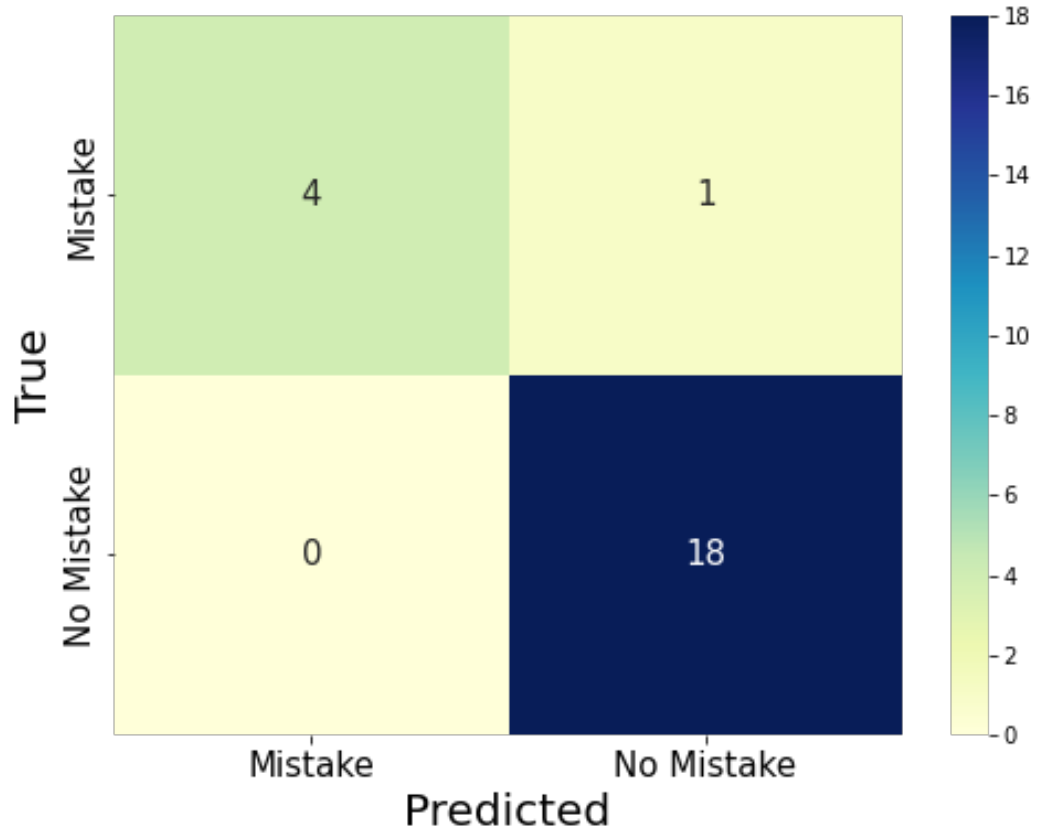

Figure 4.2: Confusion Matrix for Missing Ingredients Module

4.0.3 Wrong Measurements

The ingredient list items contain additional information such as measurements for the ingredients. This measurement is very important for reaching the desired result. If the measurement mentioned in the ingredient list does not match the total of all the references for measurement of a particular ingredient, then there is a measurement mistake in the recipe.

Out of a total of 23 recipes, there are originally 2 recipes with measurement errors, and 3 more are manually introduced to have a total of 5 recipes with measurement errors. On testing, the following results are obtained as shown in Table 4.3:

Table 4.3: Evaluation Results for Wrong Measurements Module

Number of Recipes with Mistake	**Number of Correctly Identified Mistakes**	**Number of Recipes with no Mistakes**	**Number of Incorrect Identifications**
5	4	18	0

The confusion matrix for this is shown below in Figure 4.3:

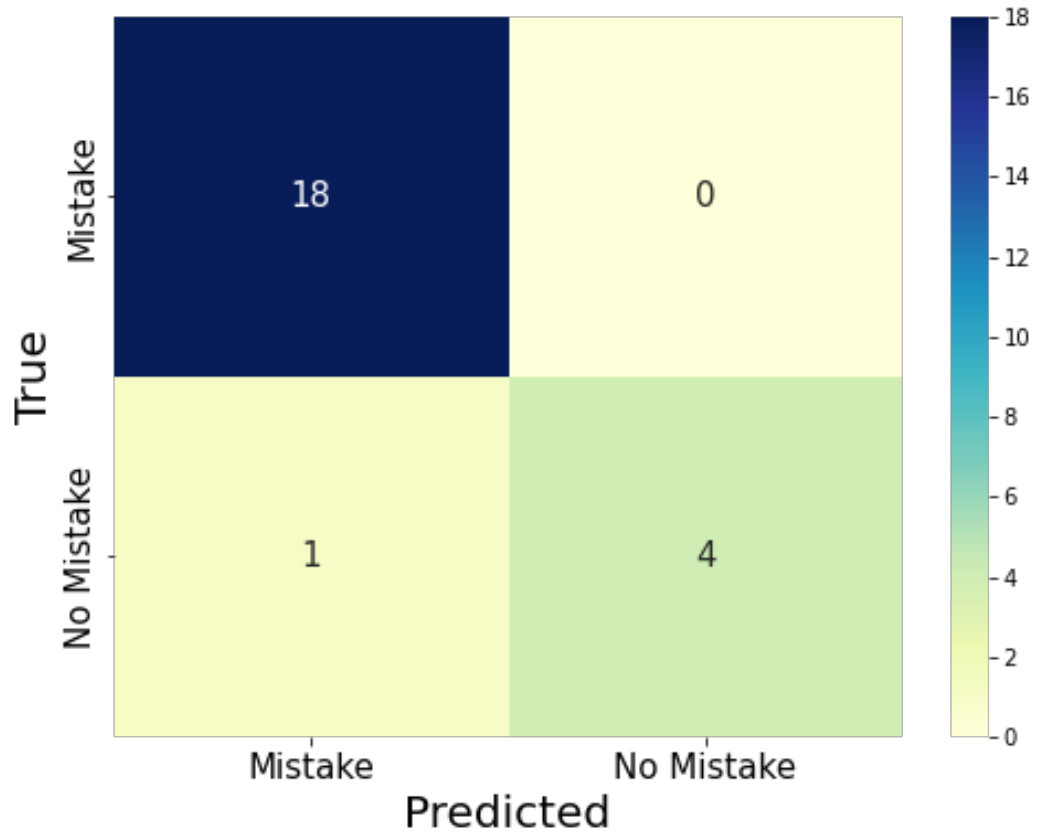

Figure 4.3: Confusion Matrix for Wrong Measurements Module

4.0.4 Incorrect Abbreviations

In cooking recipes, the units are usually referenced as their abbreviations, for example '10 teaspoons' is written as '10 tsp'. The abbreviations are sometimes mistyped which can cause confusion and errors in the measurements. If the author wants to write '10 tablespoon' the correct way is '10 tbsp' since 'tbsp' is the abbreviation for tablespoon. However, a mistype of a single alphabet and writing it as '10 tsp' will make it '10 teaspoon' which is a large difference in the measurements. Errors like this need to be identified because they can seriously affect the taste and quality of the recipe.

In the test set, there are originally no abbreviation mistakes because the original author did not use any abbreviations. 10 abbreviation mistakes are manually introduced to test the module. The results are obtained are shown in Table 4.4:

Table 4.4: Evaluation Results for Incorrect Abbreviations Module

Number of Recipes with Mistake	**Number of Correctly Identified Mistakes**	**Number of Recipes with no Mistakes**	**Number of Incorrect Identifications**
10	10	13	2

The confusion matrix for this is shown below in Figure 4.4:

4.0.5 Incorrect Cooking Temperatures and Times

Incorrect temperatures are very hard to identify for an average user of the recipes. There are some acceptable ranges for common items and if the temperature exceeds that range, then it can cause problems in the correctness of the result produced after following the recipe.

For a given temperature range, there is an acceptable cooking time range as well. If the cooking time is under this range, then the food may be left undercooked and if it is over this range, then the food may be burnt. Therefore, it is also important to identify whether the cooking time is correct or not. These two problems have to be addressed simultaneously since the cooking times directly depend on the temperature range.

In the test set, 12 temperature and time mistakes are manually introduced. The system is accurate in terms of detecting incorrect temperatures and times. The results obtained are given below in Table 4.5:

Table 4.5: Evaluation Results for Incorrect Temperatures and Times Module

Number of Recipes with Mistake	**Number of Correctly Identified Mistakes**	**Number of Recipes with no Mistakes**	**Number of Incorrect Identifications**
12	12	11	0

The confusion matrix for this is shown below in Figure 4.5:

4.0.6 Evaluation Metrics

This section will discuss the traditional machine learning evaluation metrics including Accuracy, Precision, Recall, and F1 Score for each of the modules mentioned earlier. The evaluation metrics are shown below in

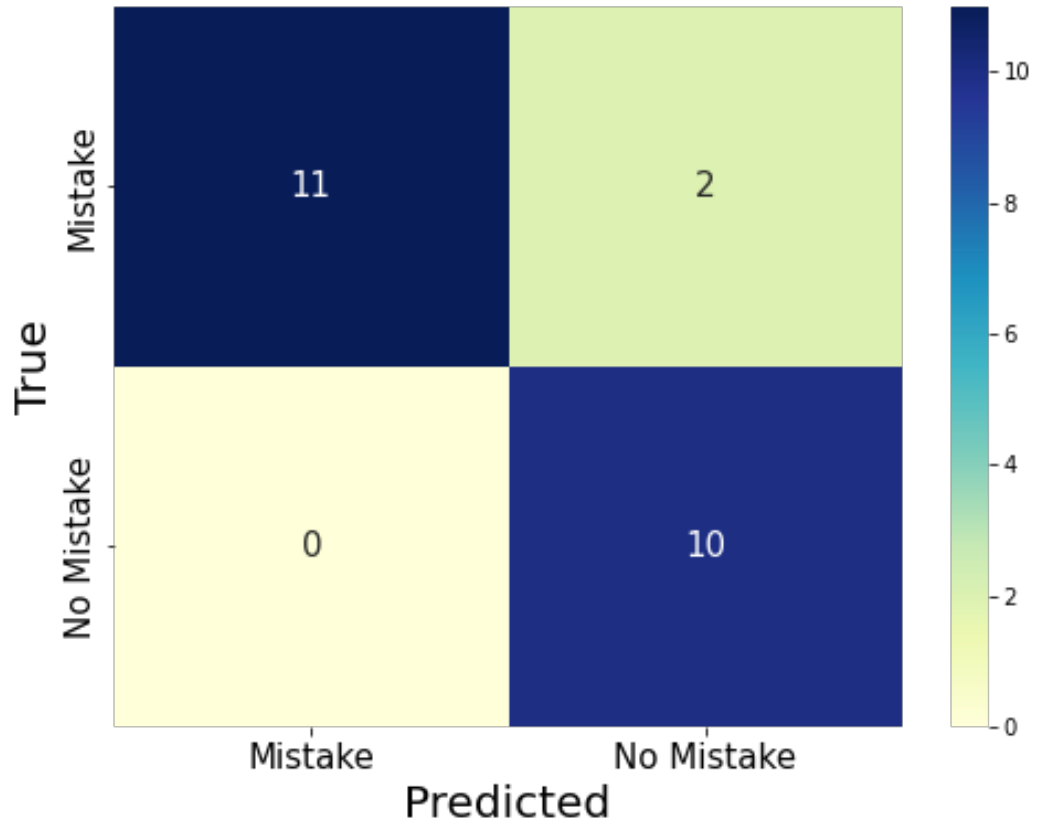

Figure 4.4: Confusion Matrix for Incorrect Abbreviations Module

Table 4.6:

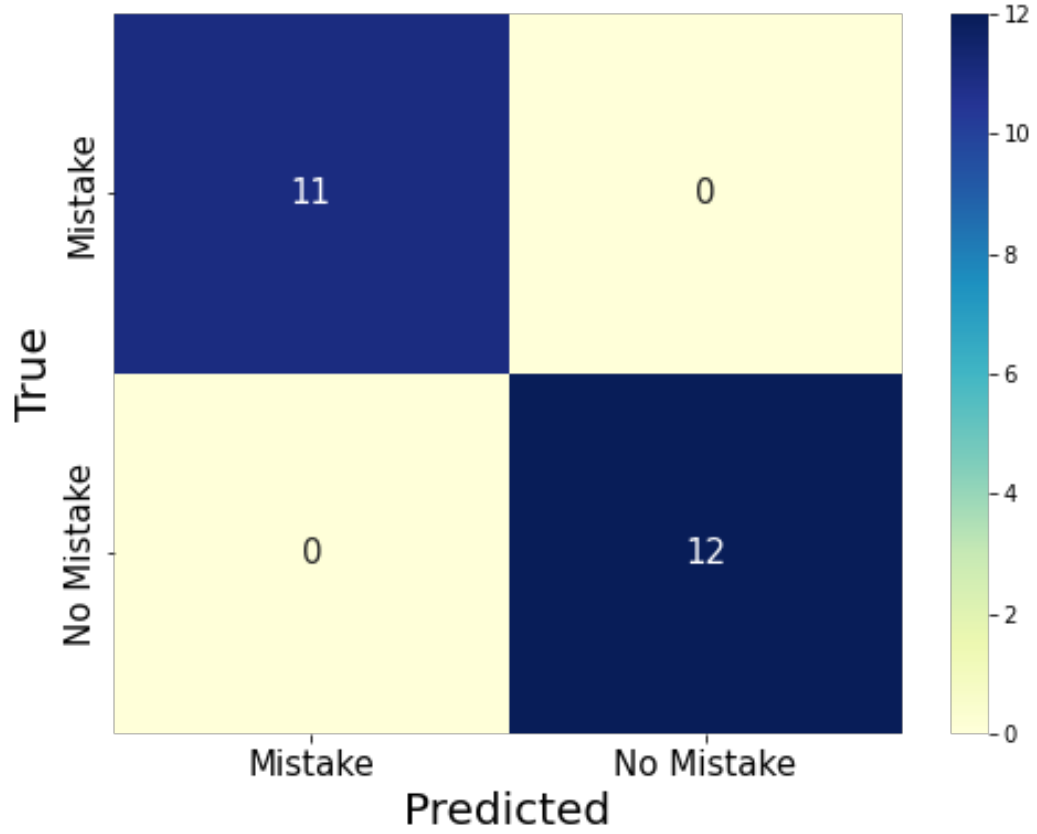

Figure 4.5: Confusion Matrix for Incorrect Temperatures and Times Module

Table 4.6: Comparison of Performance Metrics for all the modules

Module	Accuracy	Precision	Recall	F1
Out of Order Ingredients	100%	1.0	1.0	1.0
Missing Ingredients	96%	0.95	1.0	0.97
Incorrect Measurements	96%	1.0	0.80	0.89
Incorrect Abbreviations	91%	0.83	1.0	0.91
Incorrect Times and Temperatures	100%	1.0	1.0	1.0
Overall Average	**97%**	**0.96**	**0.96**	**0.95**

Chapter 5

Tools and Technologies

This chapter talks about the tools and technologies used for the development of this project. It also gives a brief rationale for the choice of a particular tool.

Programming Language: The main programming language used in this project is Python 3.9.1 [18]. Python is chosen because of the following reasons:

- Several Natural Language Processing toolkits including Spacy [19] and NLTK, etc.
- Extensive data manipulation libraries such as Pandas. [20] which is one of the most well maintained and famous open source data manipulation and analytics library.
- Ease of use with all data types including JSON which was planned to be an intermediate data type for further processing.
- Huge user community and solutions to problems faced by others which is helpful in case of problems during development.

Libraries: Several python based libraries are used for the development of this project:

- **Spacy**: Spacy [19] is used for Natural Language Processing tasks such as Named Entity Recognition and for extraction of useful information from recipe instructions. Spacy is one of the most powerful NLP libraries released so far. It is also much faster than other natural language processing libraries available in python such as NTLK.
- **Pandas**: Pandas [20] is used to develop and maintain lookup tables. It is also used to do data manipulation whenever required.
- **OS**: Python's OS library is used to interact with the operating system files whenever required.

Chapter 6

Conclusion

This project explored the research and development of a recipe analysis system that employs natural language processing for automatically detecting mistakes in the recipe instructions. During this project several interesting pieces of research on similar problems are considered to get inspiration for a methodology for developing a solution, data gathering is performed, natural language processing tools are explored to develop NER models, algorithms are developed to detect mistakes in the textual recipes and a command-line program is used to report the analysis.

The project has been a success and fulfills the requirements initially set. It can reasonably parse real-world cooking recipes, and process the instructions to detect 5 common problems in the recipes.

6.0.1 Problems Faced

During work on the project, some problems are faced. A description of the problem, the benefit is provided and the solution adopted are listed below:

- **Low Quality Data**: Initially, the problem of low quality data was not considered. Due to this, the dataset which was first collected made it difficult to analyze the recipes. One advantage of this problem is that several fixes are tried and a lot of experience is gained in trying to process the haphazard and unstructured low quality data. The problem is fixed by collecting better quality recipe data.
- **Complexity**: Some of the algorithms for finding recipe mistakes were thought to be fairly simple, but when it came to development they turned out to be very complex and required adequate focus and research to properly code them. The project in general is more complicated than originally anticipated which caused the work to be a challenging but rewarding experience.

6.0.2 Limitations

This project meets the expectations set at the start. However, with a better understanding of the problem, it is found that it has the following limitations:

- **Wide Variety of Recipes**: There is a huge difference between how the ingredient lists are formed, and how the overall text is structured among different cooking blogs on the internet. This limits the ability of the system to accurately identify cooking mistakes for all types of recipes. The system will need to be trained on a bigger dataset and will need to handle several more variations of the ingredient items list for better ingredient parsing in all situations.
- **Limited Dataset**: The system is developed on a relatively smaller dataset and it is also tested on a small dataset. There can be improvements in the performance if a better and bigger dataset is used.

- **Command Line Interface**: The system currently works using a command line interface. It can be improved by developing a web application in which users can put the recipe instructions and ingredients separately in text fields and the mistakes are displayed.

Chapter 7

Future Work

Even though the project has several limitations, it is a great starting point for developing an industry-level recipe analysis software. For future work, a large dataset can be collected with the help of cooking experts so that all the different scenarios and variations are covered to ensure the applicability of the system for different kinds of recipes on the internet. A web and mobile application can be developed for a better user experience. Advanced text processing and NLP-based deep learning techniques can be explored to improve the accuracy of the system, furthermore, some of these techniques can be combined to develop a hybrid technique that can reinforce the strengths while minimizing the weaknesses of the individual techniques.

Better algorithms can be developed for the initial cleaning of the recipe instructions. If the recipes are pre-processed optimally, it can significantly improve the system and that can be a key to transforming it into an industry-level product. However, the pre-processing and parsing of unstructured recipe data is very challenging. It can also be done by deploying the system in a real-world environment and incrementally improving the system by addressing the problems it faces when new kinds of recipes are submitted. In addition to this, using the ingredient lists and textual instructions of the recipe combined with additional metadata, it is possible to develop an advanced model to predict the tastiness of the recipe. The tastiness can be in terms of sweetness, saltiness or a general taste measures.

Appendices

Appendix A

Glossary

Machine Learning is the development of computers in which the computers can learn and be trained, so that they can perform some tasks like humnas. For example, predicting the prices of houses based on number of rooms.

Model "A (machine learning) model is a file that has been trained to recognize certain types of patterns. You train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data." [21]

Deep Learning: "Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modeling." [22]

Natural Language Processing is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language [4]

Accuracy In Machine Learning, accuracy is the percentage of the correct predictions made for a problem being solved by machine learning. It is an evaluation metric used to asses the quality of a machine learning model.

Precision, Recall and F1 Score are evaluation metrics commonly used in Artificial Intelligence (AI) applications. They are used to assess the quality of a machine learning model.

References

[1] L. Brimble, "More than 70% of adults use social media for recipes instead of cookbooks, survey finds," https://www.independent.co.uk/tech/recipes-online-cookbooks-food-inspiration-social-media-facebook-instagram-b1397624.html, 2020.

[2] TodayShow, "Editor-in-chief of food amp; wine magazine admits "i'm not a great cook"," Oct 2014. [Online]. Available: https://www.today.com/food/dana-cowins-mastering-my-mistakes-kitchen-cookbook-celebrity-chefs-share-2d80214091

[3] Wp, "Why you can't trust every recipe on the internet," Sep 2017. [Online]. Available: https://www.sunset.com/food-wine/why-you-cant-trust-every-recipe-on-the-internet

[4] "What is natural language processing?" [Online]. Available: https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html

[5] "Recipes that don't work. the unspoken problem with cookery books." [Online]. Available: https://www.matchingfoodandwine.com/news/blog/recipes-that-dont-work-the-unspoken-problem-with-cookery-books/

[6] N. Nilesh, M. Kumari, P. Hazarika, and V. Raman, "Recommendation of indian cuisine recipes based on ingredients," in *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, 2019, pp. 96–99.

[7] S. Mori, T. Sasada, Y. Yamakata, and K. Yoshino, "A machine learning approach to recipe text processing," 2012.

[8] T. Hamon and N. Grabar, "Extraction of ingredient names from recipes by combining linguistic annotations and crf selection," in *Proceedings of the 5th International Workshop on Multimedia for Cooking amp; Eating Activities*, ser. CEA '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 63–68. [Online]. Available: https://doi.org/10.1145/2506023.2506035

[9] G. Popovski, B. K. Seljak, and T. Eftimov, "A survey of named-entity recognition methods for food information extraction," *IEEE Access*, vol. 8, pp. 31 586–31 594, 2020.

[10] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," *Artificial Intelligence*, vol. 165, no. 1, pp. 91–134, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370205000366

[11] A. Bodnari, L. Deléger, T. Lavergne, A. Névéol, and P. Zweigenbaum, "A supervised named-entity extraction system for medical text," vol. 1179, 09 2013.

[12] "Once upon a chef - fresh from my kitchen to yours." [Online]. Available: https://www.onceuponachef.com/

[13] K. N. Maillard, "Recipes and cooking guides from the new york times." [Online]. Available: https://cooking.nytimes.com/

[14] Pehrsson, Haytowitz, McKillop, Moore, Finley, and Fukagawa, "Usda branded food products database: Ag data commons," Feb 2022. [Online]. Available: https://data.nal.usda.gov/dataset/usda-branded-food-products-database

[15] A. Kretser, D. Murphy, and P. Starke-Reed, "A partnership for public health: Usda branded food products database," *Journal of Food Composition and Analysis*, vol. 64, pp. 10–12, 2017, the 39th National Nutrient Databank Conference: The Future of Food and Nutrient Databases: Invention, Innovation, and Inspiration. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0889157517301734

[16] Nytimes, "Extract structured data from ingredient phrases using conditional random fields." [Online]. Available: https://github.com/nytimes/ingredient-phrase-tagger

[17] "A million news headlines." [Online]. Available: https://www.kaggle.com/therohk/million-headlines

[18] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[19] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017, to appear.

[20] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445. Austin, TX, 2010, pp. 51–56.

[21] QuinnRadich, "What is a machine learning model?" [Online]. Available: https://docs.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model

[22] "What is deep learning and how does it work?" [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network