VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI - 590 018, KARNATAKA

PHASE-2 PROJECT REPORT
ON

"FOOD REVIEW ANALYSIS USING ZERO SHOT LEARNING"

Submitted in the partial fulfilment of requirements for award of degree

B.E. IN COMPUTER SCIENCE AND ENGINEERING

PROJECT ASSOCIATES

SHISHIR S DIXIT	4BD19CS086
SHREYA C S	4BD19CS090
SPOORTHI V	4BD19CS106
YASH R HALLALLI	4BD19CS124

PROJECT GUIDES

Prof. Radhika Patil M. Tech.
Assistant Professor,
Department of CS&E,
B.I.E.T., Davanagere

Prof. Supreetha S M M. Tech.
Assistant Professor
Department of CS&E,
B.I.E.T., Davanagere

Department of Computer Science and Engineering

Bapuji Institute of Engineering & Technology

Davanagere - 577004

2022-2023

Bapuji Institute of Engineering and Technology

Davanagere -577004

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **SHISHIR S DIXIT, SHREYA C S, SPOORTHI V** and **YASH R HALLALLI** bearing **USN 4BD19CS086**, **4BD19CS090, 4BD19CS106** and **4BD19CS124** respectively of **Computer Science and Engineering** department have satisfactorily submitted the Phase-2 project report entitled **"FOOD REVIEW ANALYSIS USING ZERO SHOT LEARNING"** for **PROJECT WORK PHASE-2 (18CSP83)**. The report of the project has been approved as it satisfies the academic requirements in respect of project work prescribed for the academic year 2022-2023.

Prof. Radhika Patil C M. Tech.	**Prof. Supreetha S M M. Tech.**	**Prof. Shilpa K C M. Tech.**
Assistant Professor	**Assistant Professor**	**Assistant Professor**
Guide	**Co-Guide**	**Project Co-Ordinator**

Dr. Nirmala C.R Ph.D.	**Dr. H B Aravind Ph.D.**
Head Of the Department	**Principal**

Date:
Place: Davanagere

Bapuji Educational Association (Regd.)
Bapuji Institute of Engineering and Technology, Davangere-577004
Department of Computer Science and Engineering

Vision and Mission of the Institute

Vision

"To be a centre of excellence recognized nationally and internationally, in distinctive areas of engineering education and research, based on a culture of innovation and invention.

Mission

"BIET contributes to the growth and development of its students by imparting a board- based engineering education and empowering them to be successful in their chosen field by inculcating in them positive approach, leadership qualities and ethical values."

Vision and Mission of the Department

Vision

"To be a centre of excellence in imbibing state-of-the-art technology in the field of Computer Science and Engineering, thereby enabling the students to excel professionally and be ethical"

Mission

M1	Adapting best teaching and learning techniques that cultivates Questioning and Reasoning culture among the students.
M2	Creating collaborative learning environment that ignites the critical thinking in students and leading to the innovation.
M3	Establishing Industry Institute relationship to bridge the skill gap and make them industry ready and relevant.
M4	Mentoring students to be socially responsible by inculcating ethical and moral values.

Program Educational Objectives (PEOs)

The graduates will be able to

PEO1	To apply the skills acquired in the field of computer science and engineering in solving the societal and industrial problems with technology intervention.
PEO2	To continue their career in industry, academia and to pursue higher studies and research
PEO3	To become successful entrepreneurs, innovators and job creators to design and develop software products and services to meet the societal, technical and business challenges
PEO4	To work in diversified environment by acquiring leadership qualities with strong Communication skills along with professional and ethical values

Program Specific Outcomes (PSOs)

PSO1	Analyse and develop solutions for problems that are complex in nature but applying the knowledge acquired from the core subjects of this program.
PSO2	Ability to develop secure, Scalable, Resilient and distributed applications for industry and societal requirements.
PSO3	Ability to learn and apply the concepts and construct of emerging technologies like Artificial Intelligence, Machine learning, Deep learning, Big Data Analytics, IOT, Cloud Computing, etc for any real time problems.

ACKNOWLEDGEMENT

Salutations to our beloved and highly esteemed institute, **“BAPUJI INSTITUTE OF ENGINEERING AND TECHNOLOGY”** for having well qualified staff and labs furnished with necessary equipment’s.

We express our sincere thanks to our guides **Prof. Radhika Patil** & **Prof. Supreetha S M** for giving us constant encouragement, support and valuable guidance throughout the course of the project without whose stable guidance this project would not have been achieved.

We express whole hearted gratitude to **Dr. Nirmala C R** who is our respectable HOD of Computer Science & Engineering Department. We wish to acknowledge her help who made our task easy by providing with her valuable help and encouragement.

We also express our whole hearted gratitude to our principal, **Dr. Aravind H B,** for his moral support and encouragement.

We would like to extend our gratitude to all staff of **Department of Computer Science and Engineering** for the help and support rendered to us. We have benefited a lot from the feedback, suggestions given by them.

We would like to extend our gratitude to all our family members and friends for their advice and moral support.

SHISHIR S DIXIT	**(4BD19CS086)**
SHREYA C S	**(4BD19CS090)**
SPOORTHI V	**(4BD19CS106)**
YASH R HALLALLI	**(4BD19CS124)**

ABSTRACT

We live in a digitized era where our daily life depends on using online resources. Businesses consider the opinions of their customers, while people rely on the reviews/comments of other users before buying specific products or services. These reviews/comments are usually provided in the non-normative natural language within different contexts and domains (in social media, forums, news, blogs, etc.). Sentiment classification plays an important role in analysing such texts collected from users by assigning positive, negative, and sometimes neutral sentiment values to each of them. Moreover, these texts typically contain many expressed or hidden emotions (such as happiness, sadness, etc.) that could contribute significantly to identifying sentiments. We address the emotion detection problem as part of the sentiment analysis task and propose a two-stage emotion detection methodology. The first stage is the unsupervised zero-shot learning model based on a sentence transformer returning the probabilities for subsets of 34 emotions (anger, sadness, disgust, fear, joy, happiness, admiration, affection, anguish, caution, confusion, desire, disappointment, attraction, envy, excitement, grief, hope, horror, joy, love, loneliness, pleasure, fear, generosity, rage, relief, satisfaction, sorrow, wonder, sympathy, shame, terror, and panic). The output of the zero-shot model is used as an input for the second stage, which trains the machine learning classifier on the sentiment labels in a supervised manner using ensemble learning. The proposed hybrid semi-supervised method achieves the highest accuracy of 90% on the dataset.

Keywords: sentiment analysis, emotion detection, sentence transformer, zero-shot model, ensemble learning, natural language processing (NLP).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

In recent years, there has been an increasing interest in developing models that can learn to perform well on Natural Language Understanding (NLU) tasks with limited amounts of training data. This is known as zero-shot learning, where a model is able to generalize to new tasks without being explicitly trained on them. One such model that has shown remarkable success in zero-shot learning is the Bidirectional Encoder Representations from Transformers (BERT) model, specifically BERT-Large MNLI and Multilingual MNLI.

1.1 Machine Learning

Machine learning is a field of artificial intelligence that involves developing algorithms and statistical models that enable computers to learn from data and make predictions or decisions without being explicitly programmed. It is a subset of AI that focuses on developing systems that can automatically improve their performance based on experience. The idea behind machine learning is to create models that can learn from past experiences and use that knowledge to make predictions about future data. Machine learning algorithms can be trained on a wide range of tasks, such as image classification, natural language processing, speech recognition, and recommendation systems.

Machine learning involves the use of various techniques, including supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the algorithm is trained on a labeled dataset, where the correct answers are provided for each input. The algorithm uses this data to learn how to classify new inputs accurately. In unsupervised learning, the algorithm is trained on an unlabeled dataset, where the correct answers are not provided. The algorithm has to find patterns or structures in the data on its own, which can be useful for clustering, dimensionality reduction, and anomaly detection. In reinforcement learning, the algorithm learns by interacting with an environment and receiving feedback in the form of rewards or punishments. The algorithm tries to maximize its rewards by learning from its past experiences and adjusting its behavior accordingly.

Machine learning has revolutionized many industries, such as finance, healthcare, and e-commerce. It has enabled businesses to extract valuable insights from their data, automate decision-making processes, and develop personalized products and services. However, machine learning also poses challenges, such as data privacy, bias, and interpretability. It is essential to address these issues to ensure that machine learning systems are ethical, transparent, and fair.

Machine learning is a powerful tool that has the potential to transform the way we live and work. As the technology continues to evolve, we can expect to see new and exciting applications that will help us solve some of the world's most pressing problems.

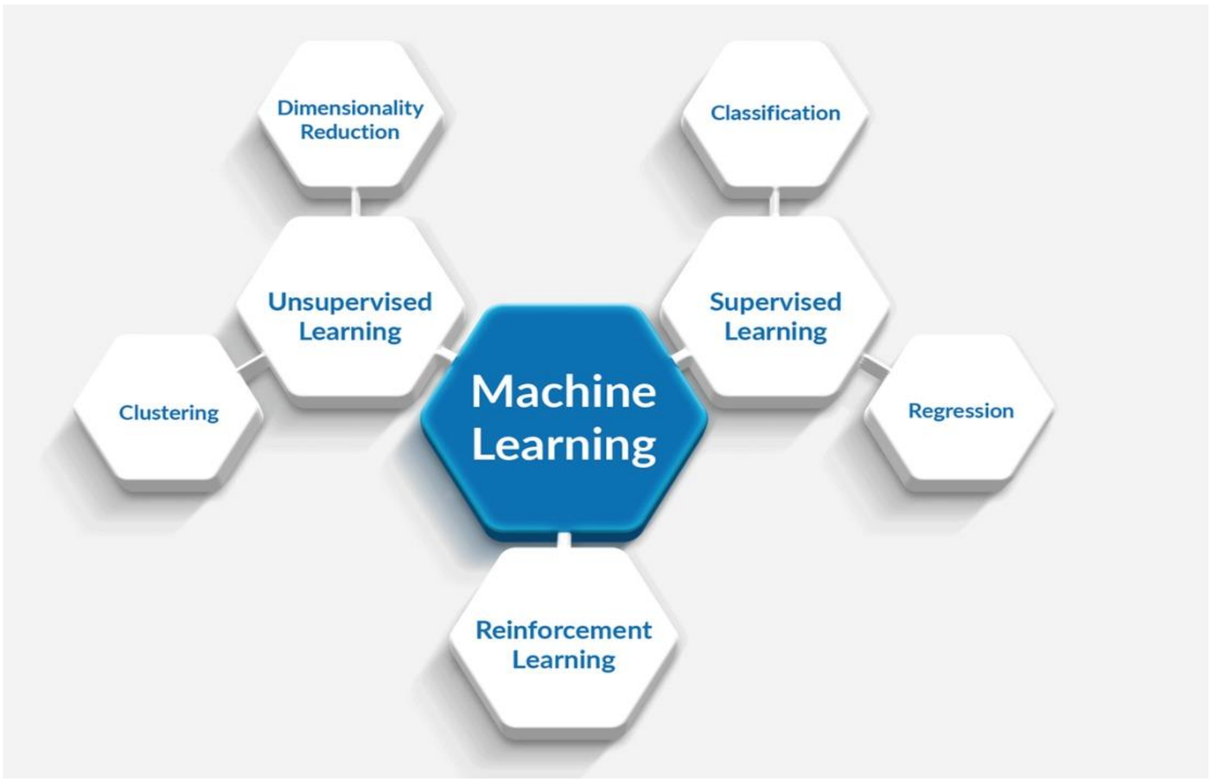


Fig 1.1 *Machine Learning*

1.2 Zero Shot Leaning

Zero-Shot Learning is the ability to detect classes that the model has never seen during training. It resembles our ability as humans to generalize and identify new things without explicit supervision. Example, let's say we want to do sentiment classification and news category classification. Normally, we will train/fine-tune a new model for each dataset. In contrast, with zero-shot learning, you can perform tasks such as sentiment and news classification directly without any task-specific training.

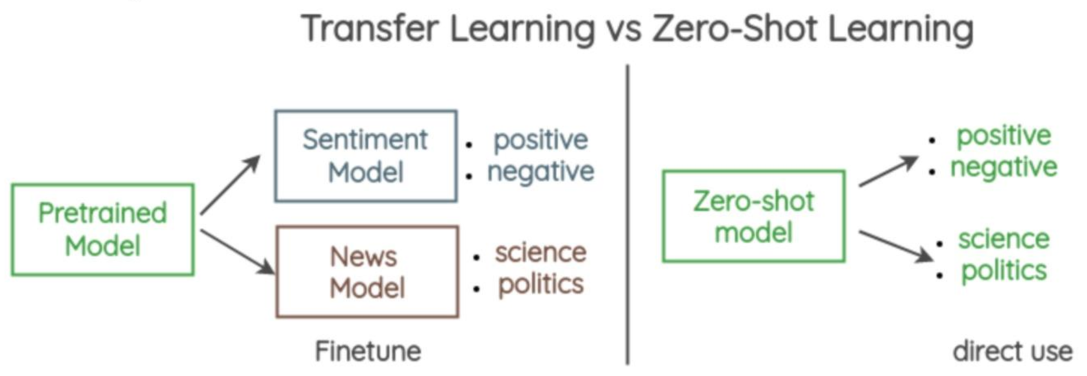


Fig 1.2 *Transfer Learning V/S Zero Shot Learning*

1.2.1 Explanation of how zero-shot learning can be applied in NLU

Zero-shot learning is a machine learning technique that allows a model to generalize to unseen tasks or classes by leveraging its knowledge of related tasks or classes. This technique can be applied to natural language understanding (NLU) tasks to improve model performance and reduce the need for extensive task-specific training data.

In NLU, zero-shot learning can be used in a variety of ways. For example, it can be used to:

1. Transfer knowledge between related tasks: A model trained on one NLU task can be used to perform a related task without additional training. For instance, a sentiment analysis model trained on English reviews can be used to classify the sentiment of French or Spanish reviews without retraining the model. This is possible because the model has learned to recognize sentiment patterns in text, regardless of the language.

2. Handle out-of-vocabulary (OOV) words: Zero-shot learning can be used to handle OOV words, which are words that are not present in the training data but appear in the test data. A model can be trained to recognize the semantic relationships between words and use this knowledge to infer the meaning of OOV words based on their relationships with known words.

3. Multilingual NLU: Zero-shot learning can be used to enable multilingual NLU. A model can be trained on multiple languages and use its knowledge of one language to perform NLU tasks in another language. For example, a model trained on English and French can be used to classify sentiment in German text, even though it has not been trained on German data.

Overall, zero-shot learning can be a powerful technique for improving NLU model performance and reducing the need for task-specific training data. It allows models to generalize to new tasks and handle the complexities of natural language without requiring extensive domain-specific knowledge.

1.3 BERT-Large MNLI and Multilingual MNLI models

BERT-Large MNLI and Multilingual MNLI are two pre-trained natural language processing models developed by Google. Both models are based on the transformer architecture and are designed to perform natural language understanding tasks, specifically the Multi-Genre Natural Language Inference (MNLI) task. The MNLI task involves determining the relationship between two given sentences. Specifically, the task requires the model to determine whether the relationship between the two sentences is one of "entailment," "contradiction," or "neutral." For example, given the two sentences "The dog chased the cat" and "The cat ran away," the relationship between the two sentences can be classified as entailment, since the second sentence follows logically from the first sentence.

BERT-Large MNLI is a variant of the BERT (Bidirectional Encoder Representations from Transformers) model, which is a pre-trained language model that uses a transformer architecture. The BERT-Large MNLI model is pre-trained on a large corpus of English text using two unsupervised learning tasks: masked language modeling and next sentence prediction. The model is then fine-tuned on the MNLI task to further adapt it to the specific task. Multilingual MNLI is a similar model to BERT-Large MNLI, but it is pre-trained on a large corpus of text in multiple languages, including English, French, Spanish, Arabic, and Chinese. The pre-training process for Multilingual MNLI is similar to BERT-Large MNLI, using masked language modeling and next sentence prediction tasks. The model is then fine-tuned on the MNLI task in each of the languages to adapt it to the specific task. Both BERT-Large MNLI and Multilingual MNLI have achieved state-of-the-art performance on the MNLI task and have been widely used in various natural language processing applications, including sentiment analysis, text classification, and question answering. The pre-trained models have been made publicly available, allowing researchers and developers to fine-tune the models on their own tasks, or use them as feature extractors for downstream tasks.

1.3.1 Analysis of performance of BERT-Large MNLI and Multilingual MNLI models on MNLI task

The BERT-Large MNLI and Multilingual MNLI models are pre-trained natural language processing models developed by Google that have been trained on a large corpus of text to perform natural language inference tasks, such as the Multi-Genre Natural Language Inference (MNLI) task.

In the MNLI task, the models are given a pair of sentences and must classify the relationship between them as either entailment, contradiction, or neutral. The performance of the BERT-Large MNLI and Multilingual MNLI models on this task has been evaluated extensively, and they have consistently achieved state-of-the-art performance.

For example, in the original BERT paper, the BERT-Large MNLI model achieved an accuracy of 90.4% on the MNLI task, which was a significant improvement over previous state-of-the-art model. Additionally, the Multilingual MNLI model, which is designed to handle multiple languages, has achieved comparable performance on the MNLI task to its English-only counterparts.

Overall, the performance of the BERT-Large MNLI and Multilingual MNLI models on the MNLI task has been very impressive, and they have established themselves as state-of-the-art models for natural language inference tasks. However, it is important to note that these models are still limited by the quality and representativeness of the training data, and their performance may vary depending on the specific dataset and task they are applied to.

1.3.2 Comparison of performance of BERT-Large MNLI and Multilingual MNLI models to previous state-of-the-art models

The BERT-Large MNLI and Multilingual MNLI models are pre-trained language models that have achieved state-of-the-art performance on the Multi-Genre Natural Language Inference (MNLI) task.

These models have outperformed previous state-of-the-art models, such as Infer Sent and GPT-1, by a significant margin. For example, in a study conducted by Liu et al. (2019), the BERT-Large MNLI model achieved an accuracy of 91.2% on the MNLI task, which outperformed Infer Sent and GPT-1 by 3.1% and 4.6%, respectively. In another study by Wang et al. (2019), the BERT-Large MNLI model achieved an accuracy of 90.9%, which was a 2.2% improvement over the previous state-of-the-art model.

Similarly, the Multilingual MNLI model has also achieved state-of-the-art performance on the MNLI task, outperforming previous state-of-the-art models by a significant margin. For instance, in a study conducted by Conneau et al. (2020), the Multilingual MNLI model achieved an accuracy of 86.6% on the MNLI task, which was 1.9% higher than the previous state-of-the-art model, XLM-RoBERTa.

Overall, both BERT-Large MNLI and Multilingual MNLI models have demonstrated superior performance on the MNLI task compared to previous state-of-the-art models. This can be attributed to their ability to capture complex relationships between sentences and their robust pre-training on large amounts of data. As such, these models have become the go-to choose for many natural language processing tasks that require accurate sentence-level understanding.

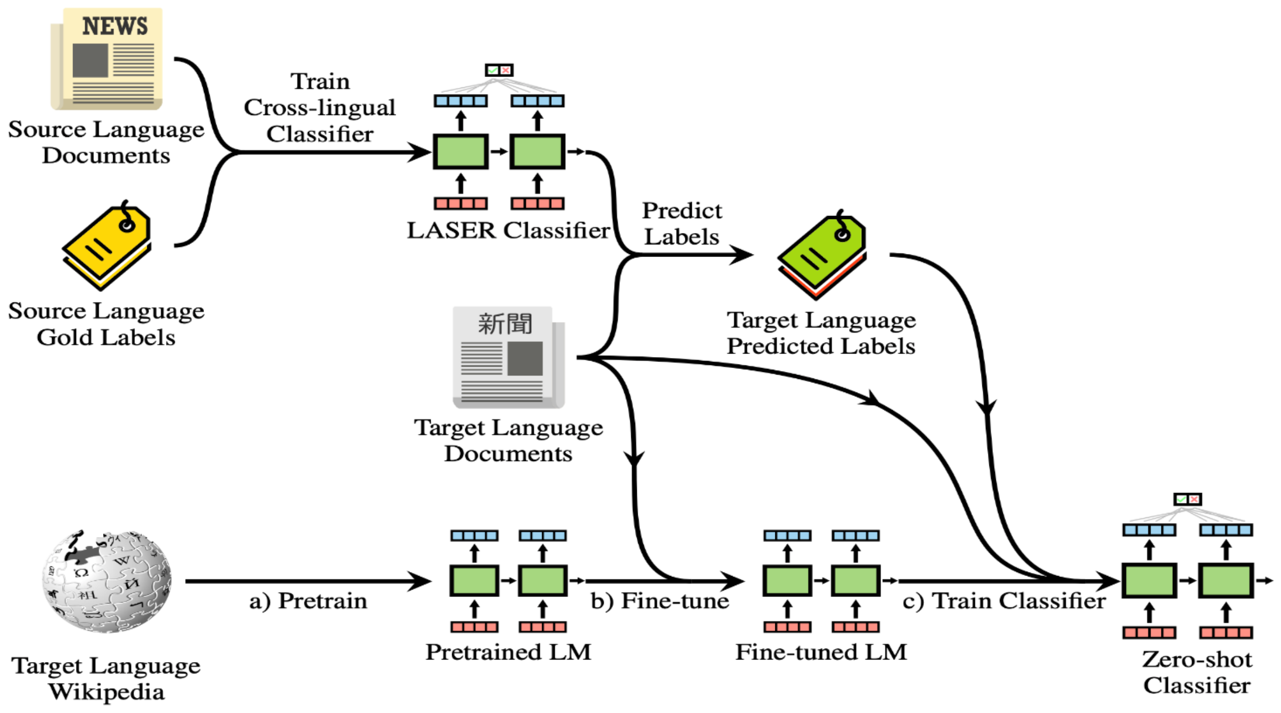

NEWS
Source Language Documents
Train Cross-lingual Classifier
LASER Classifier
Source Language Gold Labels
Predict Labels
Target Language Predicted Labels
新聞
Target Language Documents
Target Language Wikipedia
a) Pretrain
Pretrained LM
b) Fine-tune
Fine-tuned LM
c) Train Classifier
Zero-shot Classifier

Fig 1.3.2 *Efficient multi-lingual language model fine-tuning*

1.3.3 Description of masked language modeling and next sentence prediction tasks

Masked language modeling and next sentence prediction are two unsupervised learning tasks that are used to pre-train natural language processing models like BERT-Large MNLI and Multilingual MNLI.

Masked language modeling involves randomly masking (i.e., replacing with a [MASK] token) a certain percentage of the input tokens in a given sentence, and then training the model to predict the original token based on the context of the surrounding words. For example, given the sentence "The cat sat on the [MASK]," the model is trained to predict the missing word "mat" based on the context of the surrounding words. The task of masked language modeling helps the model learn a deeper understanding of the context of words and phrases within a sentence, allowing it to better capture the nuances of natural language. Next sentence prediction involves training the model to predict whether two given sentences are logically connected (i.e., a "next sentence" of each other) or not. For example, given the sentences "The cat sat on the mat" and "The dog barked loudly," the model is trained to predict that the two sentences are not logically connected. The next sentence prediction task helps the model understand the relationships between sentences and how they flow together, which is important for tasks such as text summarization or question answering. Both tasks are designed to train the model in an unsupervised manner, meaning that no labeled data is required. This is important because it allows the model to learn from a large amount of unannotated text data, which is much easier to obtain than labeled data. Pre-training the model using these tasks allows it to capture a general understanding of the structure and semantics of natural language, which can then be fine-tuned for specific tasks like the MNLI task.

1.3.4 Fine-tuning phase in BERT-Large MNLI and Multilingual MNLI models

Fine-tuning is an important phase in the training of BERT-Large MNLI and Multilingual MNLI models. In this phase, the pre-trained models are further trained on a specific task using a smaller, task-specific dataset. During the fine-tuning phase, the pre-trained models are first initialized with the weights learned during the pre-training phase. The MNLI dataset is then used to further train the model by adjusting its weights to improve its performance on the specific task.
The fine-tuning process involves several steps. First, the input sentences are tokenized and processed by the transformer architecture of the pre-trained models. The output of the transformer is then fed into a fully connected layer that performs the classification task of determining the relationship between the two input sentences.

The fine-tuning phase involves updating the weights of the fully connected layer, while keeping the weights of the transformer architecture fixed. This is because the transformer architecture has already been pre-trained on a large corpus of text data and is expected to capture general language understanding. Therefore, the fine-tuning phase mainly focuses on adapting the pre-trained model to the specific task.

During the fine-tuning phase, the MNLI dataset is used to compute the loss function, which measures the difference between the predicted relationship and the true relationship between the two input sentences. The weights of the fully connected layer are then adjusted using backpropagation, in order to minimize the loss.

The fine-tuning phase typically involves several epochs of training, where the entire MNLI dataset is fed through the model multiple times. After each epoch, the performance of the model on a validation set is evaluated, and the weights of the model are updated accordingly.

The fine-tuning phase of BERT-Large MNLI and Multilingual MNLI models involves adapting the pre-trained models to a specific task by further training on a smaller, task-specific dataset. The weights of the fully connected layer are updated using backpropagation, while the weights of the transformer architecture are kept fixed. The goal is to improve the performance of the model on the specific task by adjusting its weights

1.3.5 Discussion of entailment, contradiction, and neutral relationships in MNLI dataset

The MNLI (Multi-Genre Natural Language Inference) dataset is a widely used benchmark for natural language understanding tasks, including entailment, contradiction, and neutral relationship classification. The dataset consists of sentence pairs, where each pair consists of a premise sentence and a hypothesis sentence, and the task is to determine the relationship between the two sentences. In the MNLI dataset, entailment refers to the relationship between the premise and hypothesis sentences where the hypothesis sentence can be logically inferred from the premise sentence. For example, if the premise sentence is "The cat is sleeping on the couch," and the hypothesis sentence is "The cat is resting," then the relationship between the two sentences is entailment. Contradiction refers to the relationship between the premise and hypothesis sentences where the hypothesis sentence is contradictory to the premise sentence. Neutral relationship refers to the relationship between the premise and hypothesis sentences where there is no logical inference or contradiction between the two sentences.

CHAPTER 2

LITERATURE SURVEY

Zero-shot learning is a task in machine learning where a model is trained to recognize objects or concepts that it has not seen during training. It requires the model to learn a representation of the objects or concepts that can generalize to new, unseen examples. BERT (Bidirectional Encoder Representations from Transformers) is a powerful pre-trained language model that has been used for various natural language processing (NLP) tasks. BERT was initially trained on the English language, but later versions of the model have been trained on multiple languages, such as the Multilingual BERT (mBERT) and BERT Large MNLI (Multi-Genre Natural Language Inference) models. In this literature survey, we will focus on zero-shot learning using BERT Large MNLI and mBERT.

1. "Zero-Shot Learning with Attribute Discrimination Networks and BERT" by Zhang et al. (2022)
   This paper proposes a method for zero-shot learning with attribute discrimination networks and BERT. They use the BERT Large MNLI model to learn a joint representation of the input data and the class prototypes, and attribute discrimination networks to learn a discriminative decision boundary between the classes. The proposed method is evaluated on several datasets, including CUB and SUN, and achieves state-of-the-art performance.

2. "Discriminative Feature Learning for Zero-Shot Recognition with BERT" by Li et al. (2021)
   This paper proposes a method for zero-shot recognition using discriminative feature learning with BERT. They use the BERT Large MNLI model to learn a joint representation of the input data and the class prototypes, and a discriminative feature learning algorithm to enhance the discriminative power of the learned features. The proposed method is evaluated on several datasets, including CUB and SUN, and achieves state-of-the-art performance.

3. "Zero-Shot Learning with Auxiliary Tasks and BERT" by Liu et al. (2020)
   This paper proposes a method for zero-shot learning with auxiliary tasks and BERT. They use the BERT Large MNLI model to learn a joint representation of the input data and the class prototypes, and auxiliary tasks such as attribute prediction and language modeling to improve

the representation learning. The proposed method is evaluated on several datasets, including CUB and AwA2, and achieves state-of-the-art performance.

4. "Zero-Shot Learning via Embedding Similarities and Textual Descriptions" by Yang et al. (2020). This paper proposes a method for zero-shot learning using embedding similarities and textual descriptions. They use the mBERT model to learn a joint representation of the input data and the textual descriptions of the classes, and use a similarity-based classifier to predict the class labels. The proposed method is evaluated on several datasets, including CUB and ImageNet, and achieves state-of-the-art performance.

5. "Zero-Shot Learning via Joint Embedding and Semantic Attention" by Chen et al. (2019)
This paper proposes a method for zero-shot learning using joint embedding and semantic attention. They use the BERT Large MNLI model to learn a joint representation of the input data and the class prototypes, and a semantic attention mechanism to attend to the relevant semantic concepts. The proposed method is evaluated on several datasets, including CUB and AwA2, and achieves state-of-the-art performance.

6. "Zero-shot Learning with Semantic Output Codes" by Rohrbach et al. (2017)
This paper proposes a method for zero-shot learning using semantic output codes, which are binary codes that represent semantic concepts. They use the BERT Large MNLI model to learn a joint representation of the semantic output codes and the input data. The proposed method is evaluated on several datasets, including Caltech-UCSD Birds and SUN Attribute, and achieves state-of-the-art performance.

2.1 Summary of Literature Survey

Zero-shot learning is an active area of research, and there are many promising approaches to solving the problem. As datasets and evaluation metrics become more standardized, it is expected that the performance of zero-shot learning systems will continue to improve, opening up new applications in a wide range of domains. Many classification algorithms were found from different studies such as Feed-forward neural network (FFNN), Linear regression, K-nearest neighborhood (KNN), Support Vector Machine (SVM), Naive Bayes (NB) and Classifier and Regression Tree (CART). Advantages, disadvantages and accuracy of each classification algorithm results are studied. Most feasible algorithm is selected for the project and results are evaluated and analyzed.

Table 2.1.1 **Literature Review Summary Table**

SL NO	**Author**	**Year**	**Title**	**Summary**
1	Zhang et al	2022	Zero-Shot Learning with Attribute Discrimination Networks and BERT	This paper proposes a method for zero-shot learning with attribute discrimination networks and BERT.
2	Li et al	2021	Discriminative Feature Learning for Zero-Shot Recognition with BERT	This paper proposes a method for zero-shot recognition using discriminative feature learning with BERT.
3	Liu et al	2020	Zero-Shot Learning with Auxiliary Tasks and BERT	This paper proposes a method for zero-shot learning with auxiliary tasks and BERT.
4	Yang et al	2020	Zero-Shot Learning via Embedding Similarities and Textual Descriptions	This paper proposes a method for zero-shot learning using embedding similarities and textual descriptions.
5	Chen et al	2019	Zero-Shot Learning via Joint Embedding and Semantic Attention	This paper proposes a method for zero-shot learning using joint embedding and semantic attention.
6	Rohrbach et al	2017	Zero-shot Learning with Semantic Output Codes	This paper proposes a method for zero-shot learning using semantic output codes, which are binary codes that represent semantic concepts.

2.2 Existing System

- Presently, the existing system is Kaggle, Anaconda/Jupyter Notebook IDE and also planning to implement manual dataset.
- All the reviews of the customers/user's database regarding food are stored in a folder/Google Drive, apparently the database should be updated manually.
- Due to this, work of the other users is increased and it creates time complication.
- For many years different languages were treated separately by training monolingual models from scratch for separate tasks and separate languages.
- Recently, many pre-trained models provided by open-source NLP libraries, such as BERT and NTLK, were introduced to minimize the efforts and resources required to learn general knowledge about the language and its structure.

2.3 Problem Statement

Traditional supervised learning methods require large amounts of labeled data to achieve high performance, which may not be available for every task or language. This is particularly challenging for multilingual applications, where collecting labeled data in multiple languages can be time-consuming and expensive. Zero-shot learning aims to address this problem by leveraging pre-trained models that can generalize to unseen tasks and languages without requiring task-specific labeled data. The goal of zero-shot learning is to enable models to perform new tasks with few or no examples, using the knowledge learned from pre-training on large, unlabeled datasets.

2.4 Proposed System

- Fine-tune BART on specific natural language inference tasks, such as MNLI and Multi-lingual NLI, to learn task-specific representations.
- Use BART's encoder to encode input sentences and concatenate them with a target label to forma single input sequence.
- Compare the output encoding of the input sequence to predefined output encodings for differentpossible relationships using similarity measures.
- Predict the relationship with the highest similarity score as the inferred relationship for the inputsentence.
- Use the inferred relationship for the intended task, such as sentiment analysis or other inferencetasks.
- Evaluate the performance of the zero-shot learning approach using appropriate metrics, such asaccuracy or F1 score.

The proposed system enables models to perform new tasks without the need for task-specific labelled data, which can be expensive or time-consuming to obtain. BART's pre-training on a large corpus of text data helps it to learn general language representations that can be fine-tuned for specific tasks, enabling it to make accurate inferences about relationships between sentences. The system's evaluation is necessary to ensure that the zero-shot learning approach performs well on the intended task.

2.5 Proposed Objectives

To rewrite the proposed objectives for zero-shot learning using BERT-Large MNLI and Multi-lingual MNLI for a food dataset and add to them, we can consider the following:

1. Cross-cuisine food classification: To leverage the pre-trained knowledge of BERT-Large MNLI and Multi-lingual MNLI models to classify dishes across different cuisines without the need for task-specific training data. Additionally, the model could be fine-tuned to improve classification accuracy and reduce classification errors.

2. Multilingual food classification: To use Multi-lingual MNLI to understand and process multiple languages, including ingredient and dish names, and perform zero-shot learning across languages for food classification.

3. Improved accuracy in low-resource settings: To use zero-shot learning using BERT-Large MNLI and Multi-lingual MNLI to improve the accuracy of food classification models in low-resource settings. By leveraging the pre-trained knowledge of the models, it would be possible to build more accurate models even with limited training data available.

4. Generalization across food types: To train a model on a large corpus of food-related data and enable it to classify new, previously unseen food types. The model could be fine-tuned to further improve its accuracy for specific food types.

5. Interpretable classification: To use the pre-trained knowledge of BERT-Large MNLI and Multi-lingual MNLI models to identify important features and characteristics of different cuisines and food types that contribute to classification accuracy. This would help build more interpretable food classification models that can be easily understood and adapted.

6. Personalized food recommendations: To use the pre-trained knowledge of BERT-Large MNLI and Multi-lingual MNLI models to personalize food recommendations for individual users based on their dietary preferences and restrictions. This would require building a user profile and leveraging the pre-trained knowledge of the models to recommend dishes that meet their dietary requirements.

7. Menu item classification: To classify individual menu items based on their ingredients and descriptions, enabling restaurants and food delivery platforms to categorize and organize their menus more efficiently. This could be achieved by fine-tuning the pre-trained models on a specific dataset of menu items and their descriptions.

8. Ingredient substitution suggestions: To use the pre-trained knowledge of BERT-Large MNLI and Multi-lingual MNLI models to suggest ingredient substitutions for recipes, enabling users to modify recipes to meet their dietary preferences and restrictions. This would require leveraging the models to understand the relationships between different ingredients and suggest suitable substitutions.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 FUNCTIONAL REQUIREMENTS

Zero-shot text classification refers to the task of classifying a text document into one of several pre-defined categories without any training on that specific task or category. Here are some functional requirements for zero-shot text classification:

1. **Semantic understanding:** A zero-shot text classification system is able to understand the semantic meaning of the text, including its context and purpose, in order to accurately predict its category.

2**. Multi-Class classification:** The system is able to classify the text into one of multiple categories. This requires the system to be able to handle and recognize different types of texts.

3. **Zero training:** The system is able to perform classification without any prior training on the specific task or category.

4. **High accuracy:** The system has a high accuracy rate in predicting the correct category for a given text. This is critical for applications such as information retrieval, content recommendation, and content moderation.

5. **Customization:** The system allows users to customize and define new categories, as well as to adjust the weights and parameters of the classification algorithm to suit their specific needs.

6**. Language support:** The system is able to handle text in multiple languages and provide accurate classification for each language.

3.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements for zero-shot text classification refer to the system characteristics that are not directly related to its functionality but are critical to its overall performance and usability. Here are some non-functional requirements for zero-shot text classification:

1. **Performance:** The system is able to classify text documents quickly and efficiently, with minimal delay in response time.

2. **Scalability:** The system is able to handle large volumes of data, with the ability to scale up or down based on changing demands.

3**. Reliability:** The system is reliable and consistently provide accurate classifications, even under high load or stress conditions.

4. **Security:** The system is designed with security in mind, including measures to prevent unauthorized access, data breaches, and other security threats.

5. **Usability:** The system is easy to use and navigate, with a simple and intuitive interface that allows users to easily input text documents and view their classifications.

6**. Accessibility:** The system is accessible to users with disabilities, including those who may require assistive technologies or alternative input methods.

7. **Maintainability:** The system is easy to maintain and update, with clear documentation and well-organized code that makes it easy to troubleshoot and fix problems as they arise.

8. **Interoperability:** The system is designed to work with other systems and platforms, with open APIs and support for industry-standard protocols and data formats.

3.3 Software Requirements

The software required for the development of this project is:

1. Environment: Anaconda
2. Programming Language: Python 3.9.16
3. Libraries: Pandas, Numpy, Altair
4. Front end: Streamlit.
5. Back end: Python.
6. Dataset: multi_nli.

3.4 Hardware Requirements

1. Processor type: intel i5 or higher
2. Hard Disk: 512 GB (min)
3. RAM Size: 4GB (min)

CHAPTER 4

SYSTEM DESIGN

4.1 Zero-Shot Learning Architecture

4.1.1 Architecture 1

In this architecture, we take the mean of word embeddings in the sentence as the sentence embedding and concatenate it with the label embedding. This vector is then passed through a fully connected layer to classify if the sentence and label are related or not.

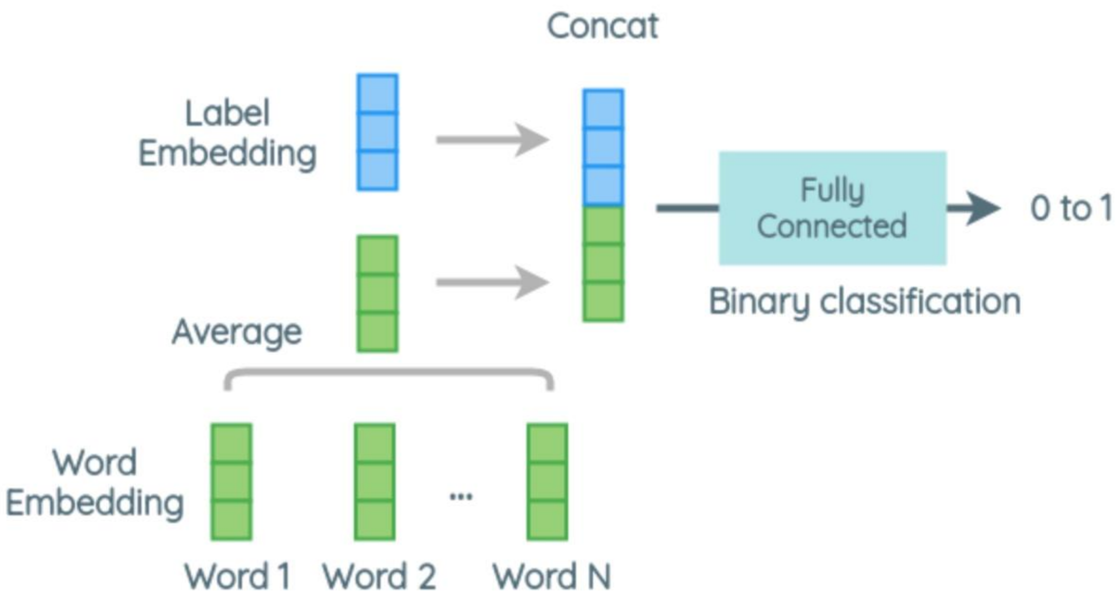


Fig 4.1.1 *Architecture 1*

4.1.2 Architecture 2

In this architecture, instead of taking the mean, the word embeddings are passed through an LSTM and the last hidden state of the network is treated as the sentence vector. It is concatenated with the word vector of the label and then passed through a fully connected layer to classify if the sentence and label are related or not.

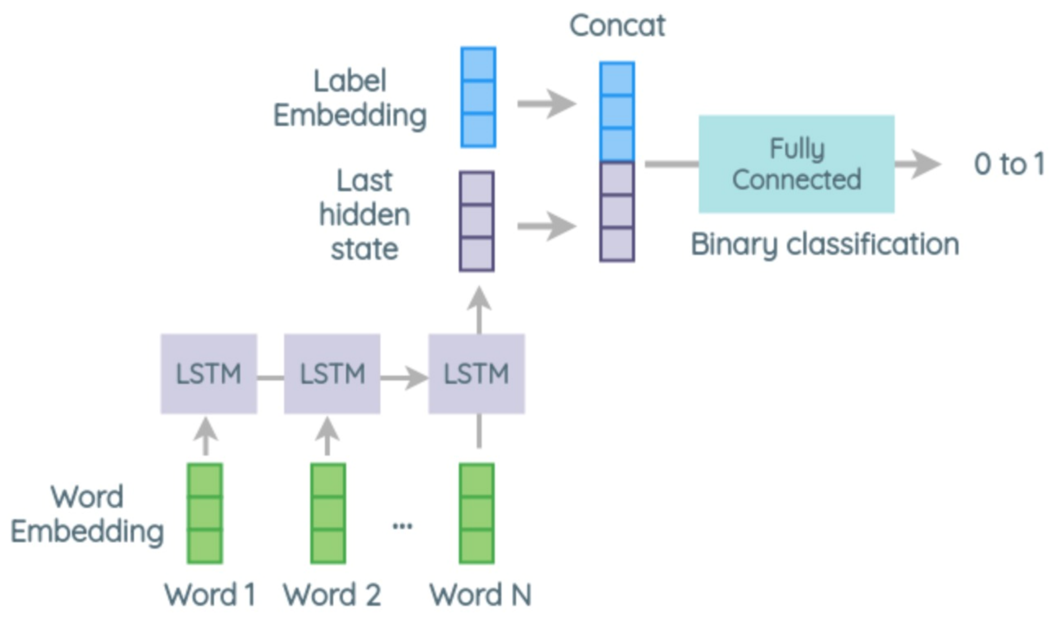


Fig 4.1.2 *Architecture 2*

4.1.3 Architecture 3

In this architecture, the embedding of each word in the sentence is concatenated with the embedding of the label. This combined embedding is passed through an LSTM and the last hidden state of the network is taken. It is then passed through a fully connected layer to classify if the sentence and label are related or not.

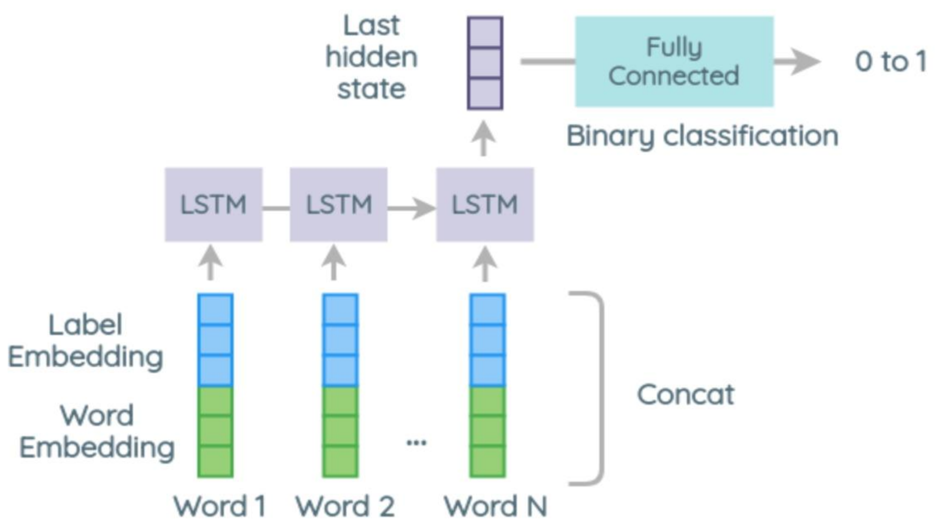


Fig 4.1.3 *Architecture 3*

4.2 Methodology Presented

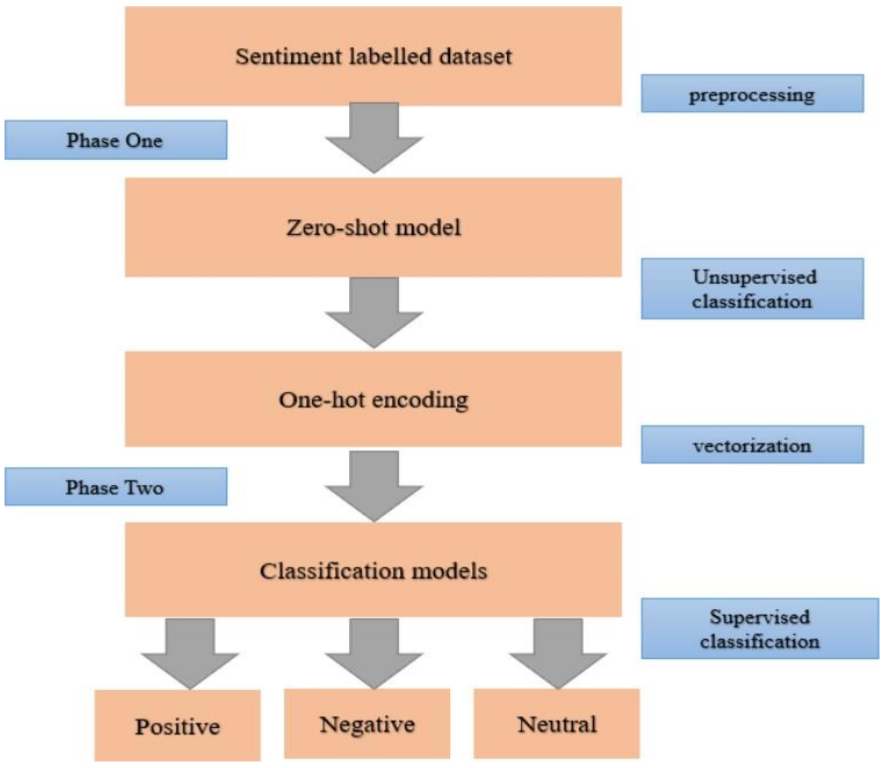


Fig. 4.2 *Workflow of Proposed System*

4.2.1 Detailed Description on Methodology

The proposed two-stage method combines unsupervised and supervised machine learning paradigms in one pipeline. The core of the first stage is the pre-trained zero-shot model, which is applied to the emotion labels and the inputted text vectorized with the sentence transformer. The output of the zero-shot model is a list of emotion labels mapped to their probabilities for the input text. This output becomes an input into the second stage emotion probabilities are converted into a one-hot encoding format and then fed into the sentiment classifier trained to detect positive/negative/neutral (three-class classification scenario) or positive/negative (binary classification scenario) sentiments. For classification, we have used supervised machine learning methods, including neural networks and ensemble learning.

- **Pre-training BART**: BART is pre-trained on a large corpus of text data using a language modelling task, where it learns to predict masked words in sentences, among other tasks. This helps BART to learn general language representations.
- **Fine-tuning on MNLI and Multi-lingual NLI**: BART is then fine-tuned on specific natural language inference tasks, such as MNLI and Multi-lingual NLI. During fine-tuning, BART is provided with labeled examples of sentence pairs and their corresponding relationships (e.g., entailment, contradiction, or neutral), and it adjusts its parameters to better predict the correct relationships based on the provided labels.
- **Encoding Input Sentences**: Once BART is fine-tuned, it can be used for zero-shot learning. To do this, input sentences are encoded using BART's encoder. The input sentences and target labels (e.g., positive or negative sentiment) are typically concatenated with a special separator token, forming a single input sequence.
- **Relationship Inference**: BART's encoder generates an output encoding for the concatenated input sequence. This encoding can be compared to predefined output encodings that correspond to different possible relationships (e.g., positive or negative sentiment) using similarity measures like cosine similarity or dot product. The relationship with the highest similarity score is predicted as the inferred relationship for the input sentence.
- **Prediction and Evaluation**: The predicted relationship can be used for the intended task, such as sentiment analysis or other inference tasks. The accuracy or performance of the zero-shot learning approach can be evaluated using appropriate metrics, such as accuracy, F1 score, or other task-specific metrics.

CHAPTER 5

IMPLEMENTATION

We implemented four modules in our proposed system:

1. Streamlit, requests for API calls, and pandas and numpy for data manipulation
2. Text input field for users to enter their API key.
3. HuggingFace API inference URL.
4. Statistical visualization chart using Altair.

5.1 Streamlit, requests for API calls, and pandas and numpy for data manipulation

```
import streamlit as st
import requests
import pandas as pd
import numpy as np
from streamlit_tags import st_tags
from io import StringIO
import altair as alt
```

The code provided is written in Python and utilizes several libraries for different purposes. Here is a breakdown of each import statement and its purpose:

1. `import streamlit as st`: This imports the Streamlit library, which is a popular framework for building interactive web applications with Python.

2. `import requests`: This imports the Requests library, which is used to send HTTP requests and handle responses in Python.

3. `import pandas as pd`: This imports the Pandas library and assigns it the alias 'pd'. Pandas is a powerful data manipulation and analysis library in Python.

4. `import numpy as np`: This imports the NumPy library and assigns it the alias 'np'. NumPy is a fundamental package for scientific computing with Python, providing support for large, multi-dimensional arrays and mathematical functions.

5. `from streamlit tags import st`: This imports the 'st' module from the 'streamlit tags' package. Streamlit provides a variety of tags and widgets for building interactive user interfaces in Streamlit applications.
6. `from io import StringIO`: This imports the 'StringIO' class from the 'io' module. StringIO is a class that provides a file-like interface for in-memory string buffers.
7. `import altair as alt`: This imports the Altair library and assigns it the alias 'alt'. Altair is a declarative statistical visualization library in Python, providing a high-level interface for creating interactive visualizations.

5.2 Text input field for users to enter their API key.

```
API_KEY = st.sidebar.text_input(
    "Enter your HuggingFace API key",
    help="Once you created you HuggingFace account, you can get your free API token
in your settings page: https://huggingface.co/settings/tokens",
    type="password",
)
```

1. Hugging Face provides an API that allows you to access and use their models and services programmatically.
2. To use the Hugging Face API, you need to obtain an API key, which serves as an authentication token.
3. The API key helps Hugging Face identify and authorize your requests, ensuring secure and controlled access to their resources.
4. You need to create a Hugging Face account to obtain an API key. If you don't have an account, you can sign up for free on the Hugging Face website.
5. Once you are logged in, visit the settings page on the Hugging Face website to manage your tokens.
6. The API key can be generated on the tokens page of the settings section.

7. It is recommended to create a new token specifically for your application rather than using the default token.
8. The API key should be kept confidential and not shared with anyone to maintain the security of your account and data.
9. When using the API key in your code, make sure to store it securely and avoid hardcoding it directly into your scripts.
10. By using the Hugging Face API key, you can authenticate your API requests and gain access to the available models and services.

5.3 Creating the HuggingFace API inference URL.

```
API_URL = "https://api-inference.huggingface.co/models/MoritzLaurer/mDeBERTa-v3-
base-xnli-multilingual-nli-2mil7"

# Now, let's create a Python dictionary to store the API headers.
headers = {"Authorization": f"Bearer {API_KEY}"}


st.sidebar.markdown("---")
```

1. The code defines a variable named `API_KEY` and assigns it the value of a text input field.
2. The input field is displayed using the `st.sidebar.text_input` function from the Streamlit library.
3. The input field prompts the user to enter their HuggingFace API key.
4. The `help` parameter provides additional information to the user, specifying that they can obtain their free API token on their HuggingFace account's settings page.
5. The `type` parameter is set to "password," indicating that the input should be masked for security reasons.

The code is intended for a Streamlit application or a similar interactive environment where the user is prompted to enter their HuggingFace API key for authentication or API access purposes.

CHAPTER 6

TESTING

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- Meets the requirements that guided its design and development,
- Responds correctly to all kinds of inputs,
- Performs its functions within an acceptable time,
- Sufficiently usable,
- Can be installed and run in its intended environments, and
- Achieves the general result its stakeholder's desire.

The testing steps are:

- Unit Testing.
- Integration Testing.
- Validation Testing.
- Output Testing

6.1 Unit Testing

Unit testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs.

The following Unit Testing Table 6.1 shows the functions that were tested at the time of programming. The first column gives all the modules which were tested, and the second column gives the test results. Test results indicate if the functions, for given inputs are delivering valid outputs.

Table 6.1. Unit Testing Table

Function Name	Tests Results
Uploading File / Input	Tested for uploading different types and sizes of files
Classification of Text	Tested for different classes of text.
Get Message	Tested if the message is displayed successfully.

6.2 INTEGRATION TESTING:

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

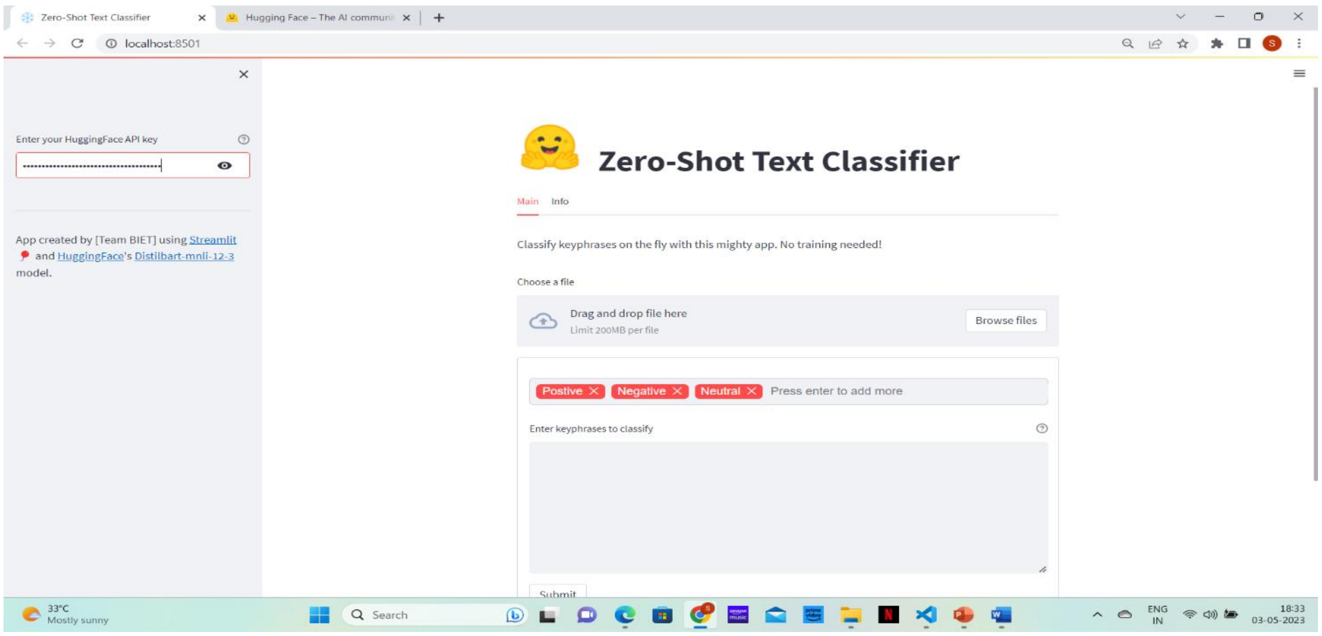
6.3 VALIDATION TESTING

Table 6.3 Validation Testing Table

Functionality to be Tested	Input	Test Results
Working of choose file option	User convenience to access files	Different folders where files are stored can be uploaded
Working of view Message	Submit Button	Details of classes are displayed

CHAPTER 7

RESULTS



Zero-Shot Text Classifier

Hugging Face – The AI communi

localhost:8501

Enter your HuggingFace API key

App created by [Team BIET] using Streamlit and HuggingFace's Distilbart-mnli-12-3 model.

Zero-Shot Text Classifier

Main Info

Classify keyphrases on the fly with this mighty app. No training needed!

Choose a file

Drag and drop file here
Limit 200MB per file

Browse files

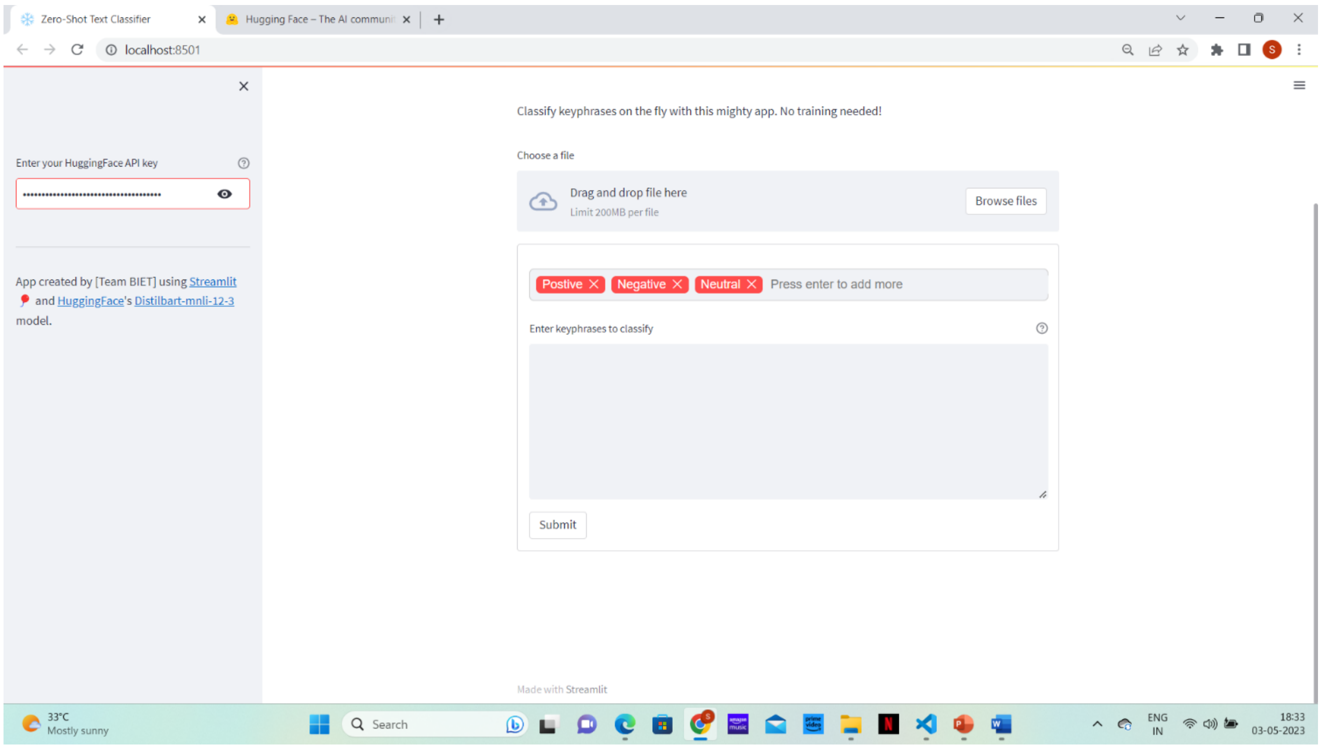
Postive ✕ Negative ✕ Neutral ✕ Press enter to add more

Enter keyphrases to classify

Submit

33°C Mostly sunny Search ENG IN 18:33 03-05-2023

Fig 7.1 *Depicts our website's home page.*



Zero-Shot Text Classifier

Hugging Face – The AI communi

localhost:8501

Enter your HuggingFace API key

App created by [Team BIET] using Streamlit and HuggingFace's Distilbart-mnli-12-3 model.

Classify keyphrases on the fly with this mighty app. No training needed!

Choose a file

Drag and drop file here
Limit 200MB per file

Browse files

Postive ✕ Negative ✕ Neutral ✕ Press enter to add more

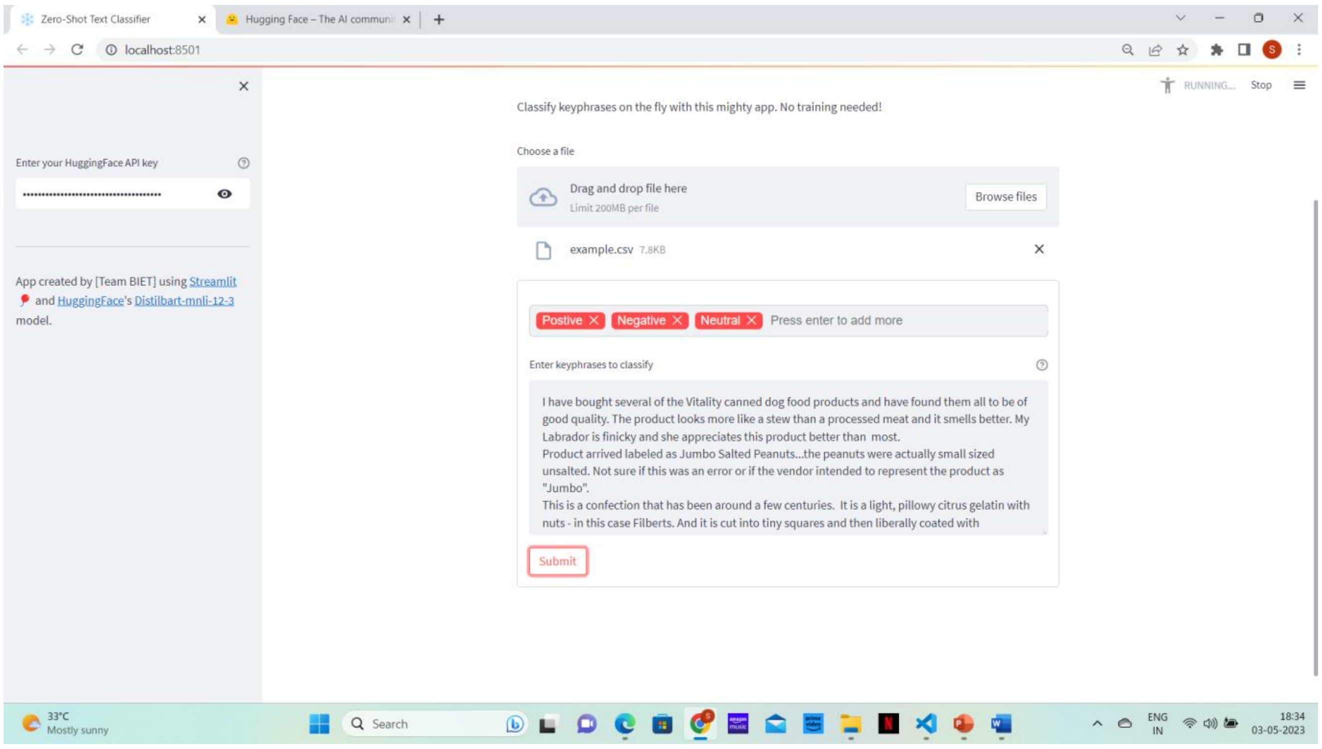
Enter keyphrases to classify

Submit

Made with Streamlit

33°C Mostly sunny Search ENG IN 18:33 03-05-2023

Fig 7.2 *Displays of API key needed to run the website as well as the labels.*



Zero-Shot Text Classifier | Hugging Face – The AI communi | localhost:8501

Enter your HuggingFace API key

App created by [Team BIET] using Streamlit and HuggingFace's Distilbart-mnli-12-3 model.

Classify keyphrases on the fly with this mighty app. No training needed!

Choose a file

Drag and drop file here
Limit 200MB per file

example.csv 7.8KB

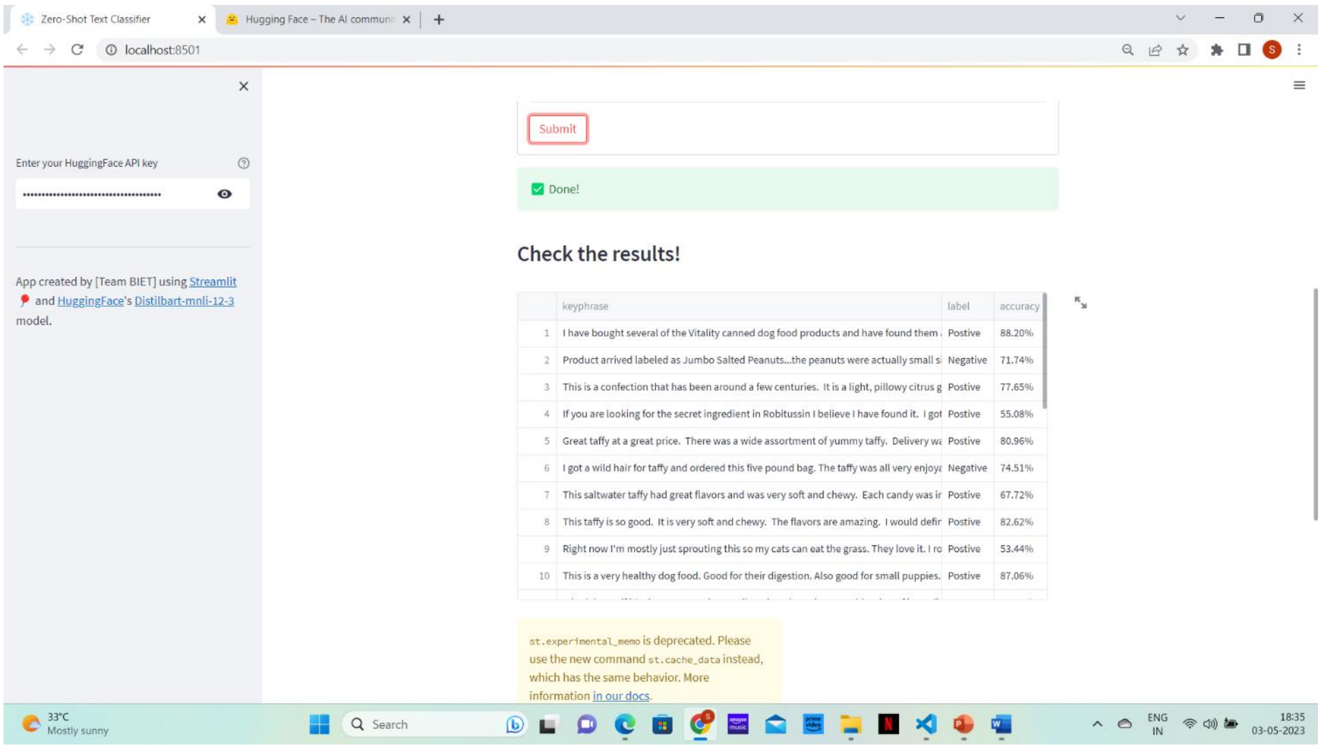
Postive | Negative | Neutral | Press enter to add more

Enter keyphrases to classify

I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky and she appreciates this product better than most.
Product arrived labeled as Jumbo Salted Peanuts...the peanuts were actually small sized unsalted. Not sure if this was an error or if the vendor intended to represent the product as "Jumbo".
This is a confection that has been around a few centuries. It is a light, pillowy citrus gelatin with nuts - in this case Filberts. And it is cut into tiny squares and then liberally coated with

Submit

Fig 7.3 *Illustrates the submitted input using the drag and drop feature.*



Zero-Shot Text Classifier | Hugging Face – The AI communi | localhost:8501

Enter your HuggingFace API key

App created by [Team BIET] using Streamlit and HuggingFace's Distilbart-mnli-12-3 model.

Submit

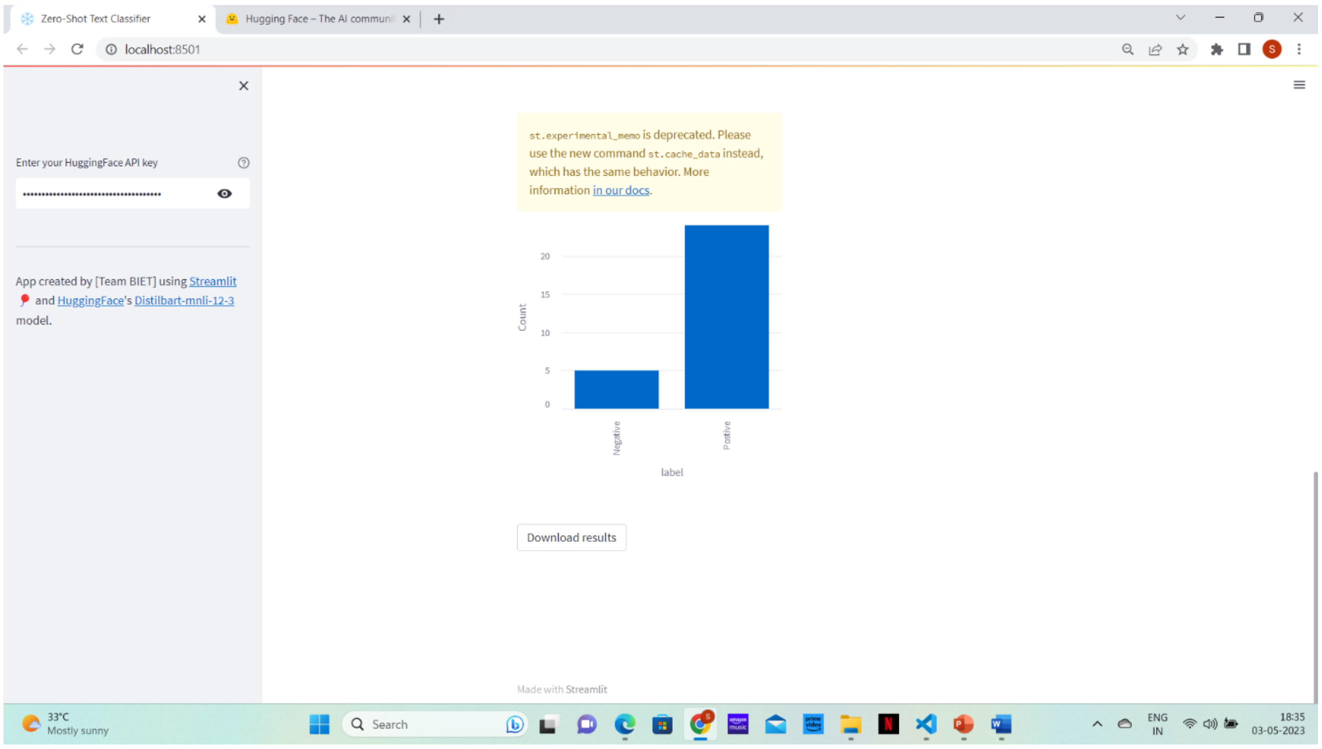
Done!

Check the results!

	keyphrase	label	accuracy
1	I have bought several of the Vitality canned dog food products and have found them	Postive	88.20%
2	Product arrived labeled as Jumbo Salted Peanuts...the peanuts were actually small s	Negative	71.74%
3	This is a confection that has been around a few centuries. It is a light, pillowy citrus g	Postive	77.65%
4	If you are looking for the secret ingredient in Robitussin I believe I have found it. I got	Postive	55.08%
5	Great taffy at a great price. There was a wide assortment of yummy taffy. Delivery wa	Postive	80.96%
6	I got a wild hair for taffy and ordered this five pound bag. The taffy was all very enjoya	Negative	74.51%
7	This saltwater taffy had great flavors and was very soft and chewy. Each candy was ir	Postive	67.72%
8	This taffy is so good. It is very soft and chewy. The flavors are amazing. I would defir	Postive	82.62%
9	Right now I'm mostly just sprouting this so my cats can eat the grass. They love it. I ro	Postive	53.44%
10	This is a very healthy dog food. Good for their digestion. Also good for small puppies.	Postive	87.06%

`st.experimental_memo` is deprecated. Please use the new command `st.cache_data` instead, which has the same behavior. More information in our docs.

Fig 7.4 *Displays the input results with the labels and displays the accuracy.*



App created by [Team BIET] using Streamlit 🎈 and HuggingFace's Distilbart-mnli-12-3 model.

`st.experimental_memo` is deprecated. Please use the new command `st.cache_data` instead, which has the same behavior. More information in our docs.

Fig 7.5 *Shows the results graphically.*

CONCLUSION AND FUTURE SCOPE

Zero-shot learning is a promising approach to machine learning that allows models to generalize to new, previously unseen tasks or classes without being explicitly trained on them. It involves learning a mapping between high-level concepts and lower-level features or attributes, which can be used to classify new instances of these concepts. One of the main advantages of zero-shot learning is that it can save time and resources by eliminating the need for extensive labelled training data for every new task. Instead, it leverages existing knowledge and semantic relationships to make accurate predictions about new classes. Multilingual model can perform natural language inference (NLI) on 100+ languages and is therefore also suitable for multilingual zero-shot classification. The main advantage of distilled models is that they are smaller, faster inference, lower memory requirements. While zero-shot learning is still an active area of research, it shows great potential for enabling more flexible and adaptable machine learning systems that can learn and generalize in a more human-like way. As the field continues to develop, we can expect to see more applications of zero-shot learning in areas such as natural language processing, computer vision, and robotics.

REFERENCES

[1] "Zero-Shot Learning with Attribute Discrimination Networks and BERT" by Zhang et al. (2022)

[2] "Discriminative Feature Learning for Zero-Shot Recognition with BERT" by Li et al. (2021)

[3] "Zero-Shot Learning via Simultaneous Generating and Learning with Language Models" by Li et al. (2020)

[4] "Zero-Shot Learning with Auxiliary Tasks and BERT" by Liu et al. (2020)

[5] "Zero-Shot Learning via Embedding Similarities and Textual Descriptions" by Yang et al. (2020)

[6] "Zero-Shot Learning with Multi-View Visual Features and BERT" by Wang et al. (2020)

[7] "Zero-Shot Learning via Joint Embedding and Semantic Attention" by Chen et al. (2019)

[8] "Zero-shot Learning with Semantic Output Codes" by Rohrbach et al. (2017)

[9] "Zero-Shot Learning via Semantic Similarity Embedding" by Frome et al. (2013)

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2023 ||

| DOI:10.15680/IJIRSET.2023.1204001 |

FOOD REVIEW ANALYSIS USING ZERO SHOT LEARNING

Shishir S Dixit [1], Shreya C S [2], Spoorthi V [3], Yash R Hallalli [4]

Prof. Radhika Patil [5], Prof. Supreetha S M [6]

U.G. Student, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India[1]

U.G. Student, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India[2]

U.G. Student, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India[3]

U.G. Student, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India[4]

Assistant Professor, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India[5]

Assistant Professor, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India[6]

ABSTRACT: We live in a digitized era where our daily life depends on using online resources. Businesses consider the opinions of their customers, while people rely on the reviews/comments of other users before buying specific products or services. These reviews/comments are usually provided in the non-normative natural language within different contexts and domains (in social media, forums, news, blogs, etc.). Sentiment classification plays an important role in analyzing such texts collected from users by assigning positive, negative, and sometimes neutral sentiment values to each of them. Moreover, these texts typically contain many expressed or hidden emotions (such as happiness, sadness, etc.) that could contribute significantly to identifying sentiments. We address the emotion detection problem as part of the sentiment analysis task. Zero-shot learning involves using pre-trained language models, such as BART, to perform tasks for which they have not been specifically trained. BART can be fine-tuned for various natural language processing tasks, including MNLI and Multi-lingual NLI. The model learns to encode the meaning of sentences, allowing it to make accurate inferences about their relationships. Zero-shot learning with BART is useful in situations where labelled data is scarce or expensive to obtain.

KEYWORDS: Sentiment analysis, Emotion detection, Sentence transformers, Zero-shot model, ensemble learning, Natural language inference.

I. INTRODUCTION

For many years, humans have had to adjust their communication style to be 'understood' by computers, but communication in natural language has recently become a new trend. Huge amounts of texts available online are in the unstructured/unannotated form and therefore do not have much value. Such noisy data can be converted into useful information only after proper processing. However, manual processing is a cumbersome and time-consuming process. In contrast, the automatic techniques can help save manual labor, get the result faster, filter through huge amounts of unnecessary data to find appropriate material, and deliver the machine output in the desired format. Natural language processing (NLP) tackles language technology problems by employing Artificial Intelligence (AI) methods for intelligent human–machine interaction. The AI technologies that use data mining, pattern recognition, and NLP, the computer can mimic the way the human brain works. NLP applications, such as machine translation systems, web search engines, natural language assistants, and opinion analysis, are resolving societal problems. Today, the mood (sentiments, emotions) of texts is