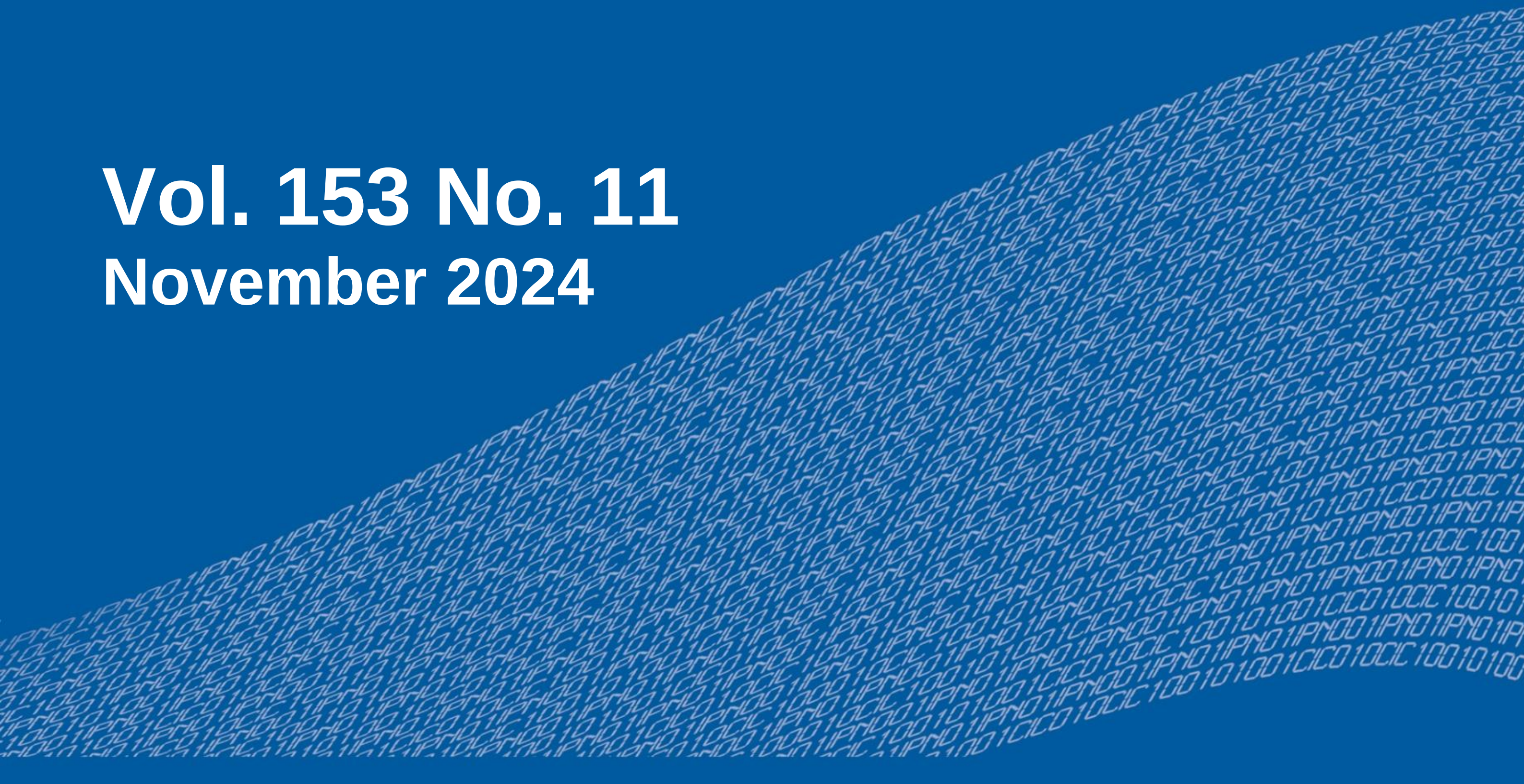
EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Research in Computing Science

Vol. 153 No. 11
November 2024



Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov, CIC-IPN, Mexico
Gerhard X. Ritter, University of Florida, USA
Jean Serra, Ecole des Mines de Paris, France
Ulises Cortés, UPC, Barcelona, Spain

Associate Editors:
Jesús Angulo, Ecole des Mines de Paris, France
Jihad El-Sana, Ben-Gurion Univ. of the Negev, Israel
Alexander Gelbukh, CIC-IPN, Mexico
Ioannis Kakadiaris, University of Houston, USA
Petros Maragos, Nat. Tech. Univ. of Athens, Greece
Julian Padget, University of Bath, UK
Mateo Valero, UPC, Barcelona, Spain
Olga Kolesnikova, ESCOM-IPN, Mexico
Rafael Guzmán, Univ. of Guanajuato, Mexico
Juan Manuel Torres Moreno, U. of Avignon, France
Miguel González-Mendoza, ITESM, Mexico

Editorial Coordination:
Griselda Franco Sánchez

Research in Computing Science, Año 23, Volumen 153, No. 11, noviembre de 2024, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de noviembre de 2024.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización del Instituto Politécnico Nacional.

Research in Computing Science, year 23, Volume 153, No. 11, November 2024, is published monthly by the Center for Computing Research of IPN.

The opinions expressed by the authors does not necessarily reflect the editor's posture.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research of the IPN.

Volume 153(11)

Advances in Artificial Intelligence

Miguel González-Mendoza (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2024

ISSN: in process

Copyright © Instituto Politécnico Nacional 2024
Formerly ISSNs: 1870-4069, 1665-9899

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Table of Contents

Page

Application of Condorcet's Jury Theorem for Enhancing Sentiment Analysis Performance Using BERT Transformers: A Case Study for Spanish 5
Gerardo Bárcena-Ruiz, Richard de Jesús Gil-Herrera

Digital Nudges within Recommendation Systems for Improving Eating Behavior towards Healthier Options: State of the Art 15
Judith Mayte Flores-Pérez, Richard de Jesús Gil-Herrera

Classification of Zika and Dengue Clinical Data Using Feature Encoding and Machine Learning Techniques 25
Elí Cruz-Parada, Guillermina Vivar-Estudillo,
Eduardo Pérez-Campos, Carlos Lastre-Domínguez

Transfer Learning: Shallow and Deep Neural Models 35
Diego Uribe, Enrique Cuan, Elisa Urquizo

Development of an Autonomous Navigation and Manipulation Robot for Obstacle-Rich Environments 49
Andrea M. Salcedo-Vázquez, José A. León-Navarro,
Hortencia A. Ramírez-Vázquez, José G. Buenaventura-Carreón,
Fernando Antunez-Arnold

Enhancing Music Genre Classification Using Tonnetz and Active Learning 63
Omar Velázquez-López, José Luis Oropeza-Rodríguez,
Gibran Fuentes-Pineda

LLAMA Assisted Nutritional Recipe Suggestions: Integrating the Dietary Quality Index for Health Conscious Cooking 75
Diego Estrada-Beltrán, Miguel Gonzalez-Mendoza,
Raul Monroy-Borja, Gilberto Ochoa-Ruiz, Janet Gutiérrez-Uribe,
Astrid Domínguez-Uscanga

Exploring Class Visualizations for Neural Architecture Search 93
Ricardo A. Zavala-Cordero, Alejandro Rosales-Pérez

ISSN 1870-4069

Application of Condorcet’s Jury Theorem for Enhancing Sentiment Analysis Performance Using BERT Transformers: A Case Study for Spanish

Gerardo Bárcena-Ruiz[1,2], Richard de Jesús Gil-Herrera[3]

[1] Universidad Americana de Europa,
Mexico

[2] Universidad Panamericana,
Mexico

[3] Universidad Internacional de la Rioja,
Spain

gbarcena@up.edu.mx, richard.dejesus@unir.net

Abstract. This paper examines the application of Condorcet's Jury Theorem (CJT) in the context of sentiment analysis using BERT models for Spanish language. While there are many BERT model variants, including vanilla BERT, DeBERTa, ALBERT or Longformer, this study focuses on BERT, RoBERTa and DistilBERT due to their superior accuracy and efficient retraining times. The objective of this research is to assess whether CJT can enhance sentiment analysis performance with BERT models. The experiments conducted explore various scenarios to evaluate the model’s behavior and the effectiveness of a jury metamodel. The CJT approach can yield superior results, achieving an F1-Score of 0.994 compared to a single model’s average F1-Score of 0.974, according to this study. Additionally, the study highlights the critical role of dataset language quality in training more effective models.

Keywords: Condorcet’s Jury Theorem (CJT), BERT, transformer, performance, sentiment analysis, Spanish language.

1 Introduction

According to Cervantes Institute [1], Spanish is the second most spoken native language by number of speakers, following Mandarin Chinese. Additionally, in a worldwide count of total speakers (including native speakers, limited proficiency speakers and students of Spanish), it ranks forth, after English, Mandarin Chinese, and Hindi.

However, research on AI involving the Spanish language occupies a relatively small market share. For instance, studies on sentiment analysis in Spanish account for only 3.11 % to 5.26% of total research in this area [2]. Natural Language Processing (NLP) encompasses a range of critical tasks heavily influenced by the target language. Core examples include Automatic Translation, Text Summarization, and Text Generation,

pp. 5–14; rec. 2024-06-04; acc. 2024-08-17

all of which necessitate effective text comprehension for optimal performance [18]. Furthermore, NLP excels in specific applications such as topic modeling (extracting semantic information), news classification (leveraging news as a rich data source), sentiment analysis (categorizing text based on sentiment), and question answering (a challenging task with potential for intuitive knowledge acquisition) [19].

In recent years, Deep Learning has emerged as the dominant paradigm for text classification [19]. The evolution of text processing methodologies progressed from Recurrent Neural Networks (RNNs) to Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures, which mitigated the vanishing gradient problem through more efficient backpropagation. However, these models faced limitations in parallelization and handling long sequences. The subsequent introduction of Transformers, equipped with the attention mechanism, revolutionized the field by addressing these shortcomings and enabling highly parallelizable text processing [18].

Transformers [20] are sequence-to-sequence models comprising encoder and decoder blocks, commonly employed in neural machine translation [21]. Bidirectional and Auto-Regressive Transformers (BART) and Text-to-Text Transfer Transformers (T5) exemplify this architecture. Disabling the encoder yields a sequence generation or language model, such as the renowned Generative Pretrained Transformer (GPT). Conversely, deactivating the decoder results in encoder-only or automatic coding Transformers, which generate input sequence representations. BERT is a prominent example of this latter category.

Given the potential of advanced text processing for the Spanish language, this study focuses on sentiment analysis utilizing BERT transformers. Based on [2], DistilBERT, BERT (vanilla), and RoBERTa were selected for their high F1-scores and efficient retraining times (F50 T50) among top-performing BERT models.

However, is it possible to enhance performance using the same models without altering their internal architecture? The principle of democracy might provide an answer to this question. If multiple entities (models) cast vote in a particular manner, the final verdict could represent the optimal value achievable. However, what is the probability to obtain correct answers? In the “Materials and methods” section, we discuss Condorcet’s Jury Theorem (CJT) and its associated probabilities, bet we can express in advance that the CJT framework employs a majority voting mechanism for jury decisions, anticipated to outperform individual judge assessments due to the collective expertise of participants in this domain.

In this study, we apply CJT as follows: a) each BERT model independently classifies every text as either positive or negative, functioning as individual judges. b) The experiment utilizes different BERT models that have an F1-Score greater than 0.5, ensuring that each model has sufficient knowledge of the dataset’s subject matter to classify the text accurately.

As a result, when BERT models perform the sentiment analysis individually, they vote on whether each text is positive or negative. According to CJT, the collective accuracy of these models, acting as jury, could potentially be higher.

1.1 General Objective

General objective is to apply the Condorcet's Jury Theorem (CJT) for enhancing the performance of different BERT transformers, which have been pretrained on from different datasets.

1.2 Specific Objectives

Subsequently, the study aims to achieve several specific objectives: i) identifying three distinct BERT models for sentiment analysis tasks in Spanish. ii) Determining the availability of datasets in Spanish suitable for training, trying to get balanced subsets containing both positive and negative sentiments. iii) Establishing a comparison framework for evaluating performance of individual models versus the application of CJT.

2 Related Works

The objective of the authors in [3] and [6] was to compare FinBERT and FinDROBERTA with GPT-4 for text classification in the financial domain, using a specially developed market-based dataset for retraining the models.

Their primary source of financial information was Bloomberg Market Wraps (BMW, spanning from 2010 to 2024). The BMW repository is a daily-consolidated summary of financial news by human journalists. The dataset examples were organized as follows: 27% neutral, 31% negative, and 42% positive headlines classifications.

The accuracies (F-score) for each model were, before retraining, as follows: 0.47 for GPT-4 and 0.44 for DistilROBERTA and FinBERT. After supervised fine-tuning (SFT), the accuracies improved to 0.51 for SFT GPT-4, 0.49 for SFT DistilROBERTA, and 0.50 for SFT FinBERT. The authors organized the models into different arrays or bags as follows:

- Bagging 1: SFT GPT + SFT DistilROBERTA + SFT FinBERT.
- Bagging 2: SFT DistilROBERTA + SFT FinBERT.
- Bagging 3: All models, SFT + No SFT.

Each bag achieved the following F-scores: 0.51 for Bagging 1, 0.52 for Bagging 2 and Bagging 3. Some important conclusions highlighted by the authors include: the bag array facilitates the learning of more sophisticated patterns; it mitigates the bias introduced by humans; it enhances the objectivity of the model training process; and the jury votes outperformed the original SFT models, although not sufficient to validate the application of CJT. An increase in model parameters does not necessarily translate to performance improvements.

The objective of study in [7] is to classify cancer types using a dense artificial neural network. To perform the classifying task, the authors used 25,000 examples from Lung and Colon Cancer Histopathological Image Dataset (LC25000), organized into five balances classes: benign lung tissue, lung adenocarcinomas, lung squamous cell carcinomas, benign colon tissue, and colon adenocarcinomas.

In the proposed algorithm, classifiers were organized into a majority voting system based on CJT as follows: after training, the best performing model for each class was saved, and the output was expressed in an array that contains votes for each class. The classification accuracy for the five classes reached 99.88% with the CJT ensemble model.

Finally, the research team in [8] attempted to classify COVID-19 using Deep Neural Networks (DNN) models and CJT. The dataset used was the Curated Dataset for COVID-19 Posterior–Anterior Chest Radiography Images (X-rays). After data cleaning, the final dataset comprised the following classes' mixture: 18.15% normal, 16.95% COVID-19, 33.07% viral pneumonia, and 31.82% bacterial pneumonia.

During experimentation, the testing accuracy values for predefined models were as follows: InceptionV3 96.45%, InceptionResNetV2 97.42%, ResNet50V2 97.90%, DenseNet121 97.75%, DenseNet201 97.26%. The accuracy values for the proposed models were the following: DETL Ensemble Model 97.26%, Jury Ensemble Model 98.22%. The results show that the CJT-based model outperformed the individual models.

While previous study has not directly employed the tools focused on in this paper, the authors demonstrate the applicability of the CJT framework to classification tasks.

3 Materials and Methods

In the following sections, we provide an explanation of the different models and datasets that were utilized to perform the experimentation.

3.1 Models

In this study, we utilized three pretrained transformer models from the Hugging Face hub: a) BERT [9], b) RoBERTa [10] and c) DistilBERT [11]. All three models have demonstrated strong performance for tasks involving the Spanish language, as evidenced by previous research [2].

For retraining purpose, we utilized Google Colaboratory Pro with T4 GPU [15], which fits to our computing needs. The parameters included a maximum length of 2048 for the tokenizer´s encoder, a batch size of thirty-two (32), and ten (10) epochs, according to which is reported in [16] this number of cycles produces a no overfitting model.

3.2 Datasets

Three datasets were employed for this experiment. The first dataset is about movie reviews expressed on the IMDB web site [12]. The remaining two datasets focus on tourism-related comments and were retrieved from Hugging Face hub: the Sepidmnorozy [13] and Alexcom [14] datasets. The IMDB and Alexcom datasets provided a balanced set of examples, equally divided between positive and negative sentiment classes. The Sepidmnorozy dataset, however, contained a slightly smaller sample size of 512 examples, maintaining a balanced distribution of positive and negative sentiment (256 each).

Table 1. Datasets, tokens characteristics.

Dataset	**Subject**	**Min**	**Max**	**Avg.**	**Std. Dev.**
IMDB	Movies	17	1251	238.319	179.927
Alexcom	Tourism	6	3153	82.531	123.599
Sepidmnorozy	Tourism	1	120	17.259	16.329

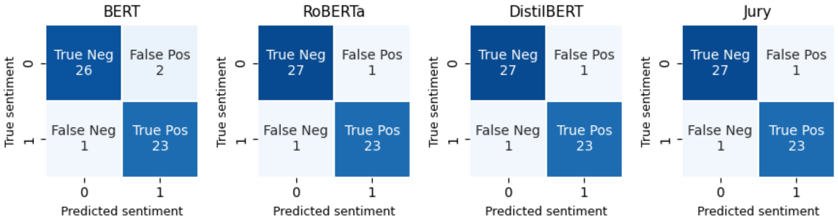


Fig. 1. Average confusion matrices for test #1.

Table 1 summarizes the tokens statistics for each dataset. As evident, the IMBD dataset exhibits a higher token count compared to the other two datasets.

3.3 Condorcet's Jury Theorem

Condorcet's Jury Theorem (CJT) [4] offers a theoretical framework [17] to address the issue of accuracy in collective decision-making tasks such as sentiment analysis. In the context of CJT, each participant can be conceptualized as a classifier, tasked with distinguishing positive from negative sentiment in text data. Each classifier must be independent, well-trained, and uniformly biased towards the correct alternative according to the following conditions:

- Independence: Each classifier must make its predictions independently.
- Identical Distribution: The performance of all classifiers should be statistically similar, implying they have the same underlying distribution of accuracy.
- Better than random: Each classifier's performance must exceed random guessing because they are well-trained.
- Uniform distribution among incorrect alternatives: All classifiers should exhibit the same probability, which is lower than random guessing, when they choose the incorrect alternative [3].

Building upon the CJT framework, we can extend its principles to the realm of ensemble classifiers. Here, CJT can be applied to a collection of classifiers, denoted as C_1, C_2, …, C_n, that satisfy the previously outlined conditions. These classifiers can be envisioned as casting votes (positive or negative sentiment) into a C_{bag}. Notably, under the CJT framework, the collective decision reached by the bag (through a majority vote) is predicted to exhibit higher accuracy compared to the individual decisions of any single classifier within the ensemble.

Furthermore, as the number of classifiers (n) approaches infinity, the probability of the C_{bag} converging on the correct sentiment classification also increases. Conversely, if the individual classifiers perform worse than random guessing, the collective decision

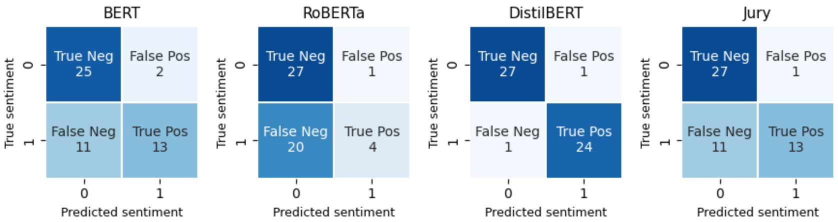


Fig. 2. Average confusion matrices for test #2.

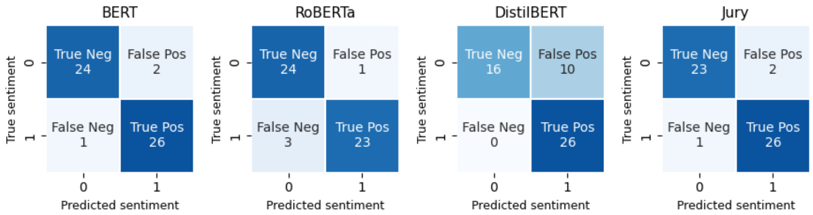


Fig. 3. Average confusion matrices for test #3.

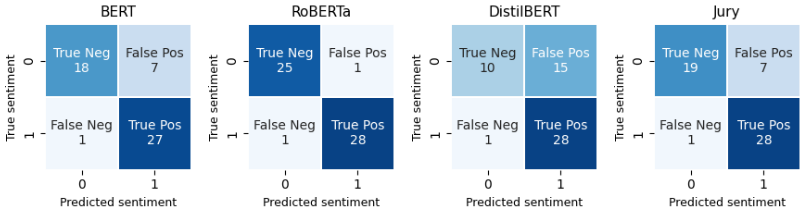


Fig. 4. Average confusion matrices for test #4.

of the C_{bag} is more likely to converge on the incorrect sentiment [3, 5]. The proof for this theorem is given in [4] and [17].

3.4 Experiment

Five ensemble models were created by combining three pretrained BERT models with available datasets as follows: BERT-IMDB, BERT-Alexcom, RoBERTa-IMDB, RoBERTa-Sepidmnorozy and DistilBERT-IMDB.

Four separate experiments were conducted, each consisting of ten rounds. During each round, 50 examples were randomly sampled from each dataset for classification (with 52 examples used from the Sepidmnorozy dataset). The summarized results of all rounds are shown in Tables 2 and Fig. 1 to Fig. 4.

4 Experimental Results and Discussion

The first experiment employed the IMDB dataset for both training and testing the ensemble models. Here, the proposed Jury (or CJT) approach achieved excellent performance, surpassing both the simple average of individual model votes and the

Table 2. Experiment statistics.

Testing Dataset		**BERT**	**RoBERTa**	**DistilBERT**	**Avg.**	**Jury**
IMDB	*Training Dataset*	*IMDB*	*IMDB*	*IMDB*		
Test	F1 Avg.	0.950	0.986	0.986	0.974	0.994
#1	F1 StdD.	0.026	0.020	0.020	0.015	0.009
	Times > Avg.	1	8	8		10
IMDB	*Training Dataset*	*Alexcom*	*Sepidmnorozy*	*IMDB*		
Test	F1 Avg.	0.754	0.596	0.990	0.780	0.788
#2	F1 StdD.	0.044	0.065	0.013	0.034	0.063
	Times > Avg.	2	0	10		5
Alexcom	*Training Dataset*	*Alexcom*	*Sepidmnorozy*	*IMDB*		
Test	F1 Avg.	0.974	0.934	0.814	0.907	0.970
#3	F1 StdD.	0.025	0.037	0.066	0.033	0.027
	Times > Avg.	10	6	1		10
Sepidmnorozy	*Training Dataset*	*Alexcom*	*Sepidmnorozy*	*IMDB*		
Test	F1 Avg.	0.862	0.994	0.707	0.854	0.879
#4	F1 StdD.	0.035	0.009	0.056	0.028	0.048
	Times > Avg.	4	10	0		5

average F1-score in 10 out of 10 rounds, according to Table 2 – Test #1. Notably, the confusion matrices exhibited perfect balance, as seen in Fig. 1.

In the second experiment, IMDB was again used for testing, while the training data comprised a combination of BERT-Alexcom, RoBERTa-Sepidmnorozy, and DistilBERT-IMDB ensembles. While BERT, RoBERTa, and the Jury (CJT) model all exhibited lower performance in this scenario, the Jury (CJT) approach still maintained a higher accuracy than the average vote and achieved a superior F1-score in 5 out of 10 rounds, as seen in Table 2 – Test #2. Interestingly, both BERT and RoBERTa models displayed a significant difficulty in identifying negative sentiment, as evident from the values in the Fig. 2, because of their high rate of False-Negative values.

The third and fourth experiments utilized Alexcom and Sepidmnorozy datasets for testing, respectively. These experiments yielded improved F1-scores. In third test, BERT achieved an F1-score of 0.974 and RoBERTa obtained a score of 0.934. Next, in fourth experiment, BERT obtained an F1-score of 0.862 and RoBERTa achieved a score of 0.994, according to Table 2. Additionally, both models BERT and RoBERTa obtained more balanced confusion matrices, as seen in Fig. 3 and Fig. 4. The Jury (CJT) model consistently achieved superior results compared to the simple average vote across all experiments.

As expected, model performance deteriorated when classifying text from a dataset that differed from the training dataset. This is evident in Test #2, where BERT achieved an F1-score of 0.754 and RoBERTa obtained a score of 0.596, as presented in Table 2. These values are lower compared to the scenario where training and testing occur on the same dataset. However, DistilBERT exhibited a contrasting behavior in Test #3. It attained an F1-score of 0.814, surpassing its performance in the fourth experiment (F1-score of 0.707).

This anomaly can be attributed to the training data used. As shown in Table 1, the IMDB dataset employed for DistilBERT's training purpose, likely contained more informative examples as evidenced by the higher average number of tokens.

5 Conclusion

In conclusion, our findings demonstrate the effectiveness of the Jury method in sentiment analysis tasks. When appropriately finetuned, the Jury approach consistently outperformed individual models. However, this success hinges on the models' exposure to domain-specific vocabulary during fine-tuning. For instance, BERT and RoBERTa exhibited strong performance when the training and testing phases involved tourism data, likely due to their familiarity with sentiment-laden tourism vocabulary. Interestingly, the DistilBERT model, fine-tuned on the movie dataset (IMDB), also achieved acceptable performance on the tourism datasets.

This can potentially be attributed to the richer vocabulary present in the IMDB dataset compared to the tourism datasets (as observed in Table 1), which provided a stronger foundation for sentiment understanding in Spanish language.

Our primary conclusion is that Condorcet's Jury Theorem can be successfully applied to sentiment analysis tasks, particularly when the constituent models possess domain-specific knowledge. This highlights the importance of tailoring models to the specific text domain being classified.

Looking towards future work, incorporating models trained on high-quality language data into the Jury framework represents a promising avenue for further improvement. This approach has the potential to deliver robust sentiment analysis across a wide range of domains.

References

1. Centro Virtual Cervantes: El español en el mundo, anuario del instituto Cervantes 2023. Spain: Instituto Cervantes (2023) https://cvc.cervantes.es/lengua/anuario/anuario_23/informes_ic/p01.htm
2. Bárcena-Ruiz, G., de-Jesús-Gil, R.: BERT transformers performance comparison for sentiment analysis: A case study in spanish. In: Rocha, Á., Adeli, H., Dzemyda, G., Moreira, F., Poniszewska-Marańda, A. (eds) Good Practices and New Perspectives in Information Systems and Technologies, WorldCIST 2024, Lecture Notes in Networks and Systems, Springer, vol. 989, pp. 152–164 (2024) doi: 10.1007/978-3-031-60227-6_13

3. Benhamou, E.: Small triumphs over large: Instances where BERT-based fine-tuned models surpass GPT-4 in classification tasks (2024)
4. Sancho, Á. R.: On the probability of the Condorcet jury theorem or the miracle of aggregation. Mathematical Social Sciences, vol. 119, pp. 41–55 (2022) doi: 10.1016/j.mathsocsci. 2022.06.002
5. Stanford Encyclopedia of Philosophy: Jury theorems. USA: Stanford University (2021) https://plato.stanford.edu/entries/jury-theorems/#CondJuryTheo
6. Lefort, B., Benhamou, E., Ohana, J. J., Guez, B., Saltiel, D., Challet, D.: When small wins big: Classification tasks where compact models outperform original GPT-4. SSRN 4780454 (2024) doi: 10.2139/ssrn.4780454
7. Srivastava, G., Chauhan, A., Pradhan, N.: CJT-DEO: Condorcet's jury theorem and differential evolution optimization based ensemble of deep neural networks for pulmonary and colorectal cancer classification. Applied Soft Computing, vol. 132, p. 109872 (2023) doi: 10.1016/j.asoc.2022.109872
8. Srivastava, G., Pradhan, N., Saini, Y.: Ensemble of deep neural networks based on Condorcet's jury theorem for screening Covid-19 and pneumonia from radiograph images. Computers in Biology and Medicine, vol. 149, p. 105979 (2022) doi: 10.1016/j.compbiomed. 2022.105979
9. Romero, M.: BETO (Spanish BERT) + Spanish SQuAD2.0. Hugging Face (2023) https://huggingface.co/mrm8488/bert-base-spanish-wwm-cased-finetuned-spa-squad2-es
10. De-la-Rosa, J., González, E., Villegas, P., González-de-Prado, P., Romero, M., Grandury, M.: Hugging Face. bertin-project/bertin-roberta-base-spanish. Hugging Face (2022) https://huggingface.co/ bertin- project/ bertin-roberta-base-spanish
11. DCCUChile: Department of computer sciences. University of Chile (2022) dccu-chile/distilbert-base-spanish-uncased-finetuned-mldoc. Hugging Face. https://huggingface. co/ dccuchile/ distilbert-base-spanish-uncased-finetuned-mldoc
12. Fernandez, L.: IMDB Dataset of 50K movie reviews (Spanish). Kaggle (2021)
13. Mollanorozy, S. sepidmnorozy/Spanish_sentiment. Hugging Face (2022) https://hugging face. co/ datasets/ sepidmnorozy/Spanish_sentiment/tree/main
14. Cortés-Miranda, I.: Hugging Face. (2023) https://huggingface.co/datasets/alexcom/analisis-sentimientos-textos-turisitcos-mx-polaridad
15. Carneiro, T., Medeiros-Da-Nóbrega, R. V., Nepomuceno, T., Gui-Bin, B., De-Albuquerque, V. H. C., Rebouças-Filho, P. P.: Performance analysis of google colaboratory as a tool for accelerating deep learning applications. IEEE Access, vol. 6, pp. 61677–61685 (2018) doi: 10.1109/ACCESS.2018.2874767
16. Komatsuzaki, A.: One epoch is all you need. (2019) http://arxiv.org/abs/1906.06669
17. Berend, D., Paroush, J.: When is Condorcet's jury theorem valid? Social Choice and Welfare, vol. 15, no. 4, pp. 481–488 (1998) doi: 10.1007/s003550050118
18. Gillioz, A., Casas, J., Mugellini, E., Abou-Khaled, O.: Overview of the transformer-based models for NLP tasks. In: 2020 15th Conference on computer science and information systems (FedCSIS), pp. 179–183 (2020) doi: 10.15439/2020F20
19. Dogra, V., Verma, S., Kavita, Chatterjee, P., Shafi, J., Choi, J., Ijaz, M. F.: A complete process of text classification system using state-of-the-art NLP models. Computational Intelligence and Neuroscience, vol. 2022, no. 1, p. 1883698 (2022) doi: 10.1155/2022/ 1883698
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I.: Attention is all you need. In: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, pp. 1-11 (2017)

21. Lin, T., Wang, Y., Liu, X., Qiu, X.: A survey of transformers. AI Open, vol. 3, pp. 111–132 (2022) doi: 10.1016/j.aiopen.2022.10.001

ISSN 1870-4069

Digital Nudges within Recommendation Systems for Improving Eating Behavior towards Healthier Options: State of the Art

Judith Mayte Flores-Pérez[1], Richard de Jesús Gil-Herrera[2]

[1] La Universidad Americana de Europa UNADE,
Mexico

[2] Universidad Internacional de La Rioja (UNIR),
Spain

`mayte_fp@cuautitlan.unam.mx, richard.dejesus@unir.net`

Abstract. Digital nudges are subtle interventions designed to influence people's behavior in a non-intrusive way, through elements such as reminders, suggestions or strategic incentives. In the context of healthy food recommendation systems (FRS), these nudges are used to encourage more health-beneficial choices, such as highlighting foods that are low in calories, rich in nutrients, or with specific properties, highlighting their health benefits, and provide personalized Recommendations based on user preferences. This document presents the results of research based on the analysis of the state of the art, synthesizing the most relevant findings on digital nudges and recommendation systems in the field of healthy eating. The Systematic Literature Review (PRISMA-ScR) methodology was used, which made it possible to identify trends, challenges and opportunities in this area. Among the main conclusions, the importance of developing and implementing new healthy recommendation systems that combine the use of digital nudges with advanced artificial intelligence technologies stands out. This approach promises to personalize recommendations, improve user experience, and have a positive impact on promoting healthy eating habits. Furthermore, the need to adapt these tools to various cultural and social contexts to maximize their effectiveness is highlighted.

Keywords: Digital nudges, eating behavior, recommendation systems, artificial intelligence.

1 Introduction

Poor eating habits contribute to affecting our long-term health. If poor eating habits are established from a very young age, they can last into adulthood, which can increase the risk of developing non-communicable diseases such as obesity, diabetes, and hypertension [1]. The correct correlation between nutrients, foods and dietary patterns can be very relevant in terms of prevention of the diseases mentioned above [2].

The food choices made throughout life are mainly influenced by family, peers and social environment [3]. Typically, parents provide food to eat according to their

Table 1. A systematic review of the literature**.**

Research Questions	What are the most used techniques in health and nutrition recommendation systems? What Artificial Intelligence (AI) tools, methods and algorithms are useful in recommendation systems in the field of healthy eating? What are the types of pushing techniques that are most relevant in these domains?
Protocolo de búsqueda	
Search string	"Health recommendation systems", "Healthy eating recommender systems" or "Healthy food recommender systems through digital nudges".
Metadata Search	Title, Abstract, Keywords
Selecting Libraries	PubMed, Google Scholar, Scopus y ScienceDirect.
Criterios de selección	
Inclusion criteria	The work must have been published within the last 5 years. The work includes documentation on aspects related to health and nutrition recommendation systems. The article describes the characteristics and main AI techniques of health and nutrition recommendation systems. Describe aspects related to the outstanding techniques of digital push in healthy food recommendation systems. Presentations in English and Spanish. Articles, publications in conferences, theses and in indexed journals.
Exclusion Criteria	The work is not recent. The work is not related to the domain. Only the summary is displayed.

preferences and experiences, regardless of age, gender, socioeconomic status, country, etc. [4]. Despite this, studies show that eating behavior that is modified during childhood can promote health [1]. Currently, there is a large amount of information on the internet that can apparently be useful to improve eating habits, there are websites that provide information about the nutrients of food, shared diets, and the like; However, in many cases, this information is not usually adapted to the nutritional needs or preferences of the users.

There are software systems that specialize in this domain, such as recommendation systems (SRs) which, according to [5], are defined as software designed to interact with large volumes of information, but which only provide the elements that are relevant to the user. These SRs are used in many areas, such as health and nutrition. Modern SRs use algorithms and models from Artificial Intelligence and machine learning, among other techniques, to analyze ingredients, based on various factors such as dietary restrictions, ingredient compatibility, and user preferences [6], to provide them with much healthier suggestions and avoid long-term health problems.

Today's technologies are providing a new landscape that allows us to intervene in improving eating behavior and habits in support of classic recommender systems. Such as digital nudges, which are elements of the user interface that allow certain options to be established in digital environments to channel users' choices, without interfering with their freedom of choice [7]. In recent years, this novel technology has been reinforced as a strategy to promote the selection of healthier products [8], being used as a healthy choice strategy within food recommendation systems (FRS).

The main objective of this research is to synthesize the main findings about digital nudges and their use within SRs to improve eating behavior towards healthier options.

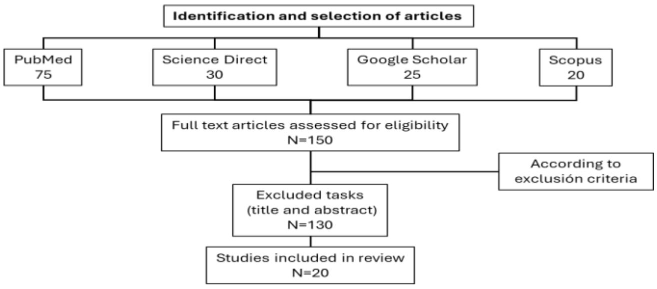

Identification and selection of articles
PubMed 75
Science Direct 30
Google Scholar 25
Scopus 20
Full text articles assessed for eligibility N=150
According to exclusión criteria
Excluded tasks (title and abstract) N=130
Studies included in review N=20

Fig. 1. Item Identification and selection.

Table 2. Search strings.

Research Question	ID	String
What are the most used techniques in health and nutrition recommendation systems?	1	HRS Tools
What Artificial Intelligence (AI) tools, methods and algorithms are useful in recommendation systems in the field of healthy eating?	2	AI tools, methods, and algorithms
What are the types of nudging techniques that are most relevant in the selected domain?	3	RS and digital nudges

Table 3. Classification of selected articles.

ID	PubMed	Science Direct	Google Scholar	Scopus
1	4	1	0	0
2	1	1	3	1
3	3	2	1	3

Likewise, the specific objectives are the three: 1) To select from the technical literature those associated with the central theme. 2) Apply inclusion and exclusion criteria according to the literature review methodology and systematic mapping based on the PRISMA-ScR protocol. [9], 3) To synthesize the findings, according to authors, reference, year of publication, contribution and techniques used.

2 Methodology

Following the PRISMA-ScR protocol, Preferred Reporting Items for Systematic Reviews of Meta-Analysis for Scoping Review (PRISMA-ScR), we developed our Systematic Literature Review, which allows mapping the existing literature related to the implementation of digital nudges and Intelligence tools. in the field of health, especially in recommendation systems focused on nutrition, as a strategy to improve

Table 4. Recommendation systems in the field of health and nutrition.

Authors	Contributions	Basic HRS Techniques
Abhar et. al, 2019. [10]	They conducted a systematic review based on PRISMA and found that hybrid and knowledge-based techniques are more widely used.	Hybrid and knowledge-based techniques.
Rya, Hyu-ho, Won-jin & Jae-dong,2019.[11]	They proposed an intelligent SR based on hybrid learning methods.	Hybrid models and machine learning.
Cai, Yu, Kumar Gladney &Mus-tafa. 2021. [12]	They focus on five aspects: The field of health, nutrition, recommended elements, recommendation technique and evaluation.	Knowledge-based approach.
Sun, Zhou, Ji,Pei and Wang, 2023. [13]	They provided 23 studies, where they reported that hybrid SRs are the most widely used in the healthcare field and the most used interfaces are usually mobile apps.	Collaborative filtering; Content-based filtering; Knowledge-based filtering; SR hybrids.
Year, Alzahrani, Martínez &Ro-dríguez 2023 [1]	A study on FRS for diabetic patients was presented within the framework of PRISMA 2020	Content-based recommendations; collaborative filtering, hybrid recommendation.

people's eating behavior, towards healthier options. This methodology consists of the following phases:

Phase 1. Planning the Review

In this phase, the title, abstract and keywords of various works related to the domain of this work were reviewed, within the main libraries selected (PubMed, Science Direct, Google Scholar and Scopus), as the most used databases currently.

Establish of research questions:

Because this work aims to find current evidence that digital nudges are useful in improving eating behavior towards healthier choices within SRs, the research questions are: 1) What are the most commonly used techniques in recommendation systems in the field of Health and Nutrition? 2) What Artificial Intelligence (AI) tools, methods and algorithms are useful in recommendation systems in the field of healthy eating? 3) What are the types of pushing techniques that are most relevant in these domains?

Protocol Review: Following the research questions of the previous phase, a search strategy was developed in the libraries, using the keywords: "Health recommendation systems", "Healthy eating recommendation systems", "Digital nudges", "Digital nudges in food recommending systems" or "Recommendation system with digital nudges". The search retrieved articles from 2018 to date.

Validation of the revised protocol: To validate the review protocol for each article, it was developed under certain selection and quality criteria. Articles were eligible to be included in this paper if: their publication was within the period range, their information was related to healthy food recommendation systems, their content has information related to digital nudges and artificial intelligence tools, they are in English or Spanish, and if their paper is complete and contains the abstract. On the other hand,

Table 5. AI algorithms and technological tools used in nutrition recommendation systems.

Authors	Contributions	Techniques & algorithms
Ganju, Tan, et al. & Me nezes 2020. [2]	They created the Saathealth Android mobile app that provides low-income families in India with health and nutrition content, which evolves thanks to scalable predictive models like Random Forest.	Machine learning. Logistic Regression Methods, Decision Trees, Random Forest
Chava,Thoms& Isaacs, 2021[5]	They adopt the standard cross-industry process for data mining (CRISP-DM), to create a recommendation system.	Collaborative content-based filtering, hybrid recommendation model.
Yi-Ying, Su-cheng, Pal-anisamy, El-ham Abdul-wahab & Hairulnizam, 2023. [6]	They point out that FRS are generally classified into 4 types: the one of users' previous preferences, the one of nutritional needs based on their health conditions, the one that focuses on balancing nutritional needs and preferences, and the one that recommends foods for certain users or groups.	Collaborative filtering, AI techniques: Matrix factorization algorithm, classification machine learning algorithms.
Abhari et al.,2023 [3]	They created a healthy food recommendation system for Iran, implemented as a website and mobile app.	Content-based, collaborative filtering, knowledge-based, hybrid. & AI Techniques: K-means, rules-based ontologies
Chen, Guo, Fan, Dong & Dong, 2023 [4]	They performed a consistent health FRS, using knowledge graphs and multitasking learning in a convolutional neural network.	Score-based method, collaborative recipe knowledge graph using TransD. AI techniques: convolutional neural network
Forouzandeh, Rostami, Berahmand &Sheikhpour, 2024[5]	They developed an HFRS-DA framework, a healthy recipe SR with heterogeneous information networks that establish links between users and recipes based on shared ingredients and nutritional content.	AI Techniques: Graphic neural networks. Deep learning. Convolutional neural networks.

the exclusion criteria for each article were: the recommendation system was not related to human health, whether the recommendations did not contain AI technological tools or contained digital nudges. Table 1 shows a summary of the information obtained in phase 1 of this methodology.

Phase 2. Conducting the Review

For each article found, a review was carried out at the abstract level, the full text and the keywords and, following the proposed inclusion and exclusion criteria, all the selected citations, the documents that successfully met them were added to Mendeley Reference Manager. To this end, the following tasks were carried out:

Identification of sources: The largest possible number of primary studies related to the research questions were identified and it was verified that the documents were in the selected period, with good spelling, with all their bibliographic sources, that they were accessible on the web and that their hypotheses were relevant to the topic addressed. In this activity, around 150 documents were found.

Selection of primary studies: After reviewing all primary studies, those that met the inclusion and exclusion criteria described in phase.

Table 6. Digital nudges techniques in recommendation systems.

Authors	Contributions	Digital nudges techniques
Karlsen & Andersen, 2019 [6]	They compare the digital nudge with HRS models.	Personalized and transparent nudge.
Dolgopolova, Toscano & Roosen 2021 [7]	Results of an experiment to encourage young people to reduce calories in fast food orders are presented.	Ordering nudges.
Starke, Majjodi & Trattner, 2022 [8]	In their study, they found that placing nutrition labels in the shape of a traffic light, along with an explanation, helps users make healthier choices	Traffic light labels to support offline choice.
Majjodi, Starke & Trattner. 2022.[9]	They conducted a study with 6 different recommendation interfaces with personalized and non-personalized recipe tips and 3 labels: no label, multiple traffic light, and nutritional score. They reveal that the digital nudge reduces difficulty of choice.	Multiple pushes of traffic lights.
Yi, Kanetkar & Brauer, 2022 [10]	They designed and implemented 5 interventions to promote fruits and vegetables in university canteens implementing Digital Nudges.	Selection of Architecture Techniques.
Castiglia, et. al. 2023 [11]	They examine the influence of multimodal interactions on a conversational food recommendation system, obtaining as a result that the chatbot is more effective when accompanied by nutritional labels.	Forms of interaction: exact, multimodal (text and images), and multimodal complemented by nutritional labels.
Ytreberg, Alfnes and Oort. 2023 [12]	Based on the taxonomy of Münscher et al. (2016) and nudges, they propose three choice architecture techniques: Decision Information, Decision Structure, and Decision Assistance.	Options Architecture Techniques: Decision information, decision structure and assistance. The decision structure is the most used technique, as a promotion of products.
Mazzori, Starke, Elahi & Tratner, 2023 [`13]	They created an experiment for a SR Online Prescriptions	Cognitive Digital Nudges (Food Labels)
Chiam et al., 2024. [14]	They designed and implemented an AI-powered platform for digital algorithms and nudges, enabled by a graph neural network (GNN)-based recommendation and granular health behavior data from wearable fitness devices	Personalized pushes that adapt to the behavior and preferences of the participants. Gamification, Reminder and social influence.

Evaluation of the quality of the studies: The process of selection and evaluation of the works was carried out by a doctoral researcher, a doctoral student and us. We only assess the quality of the studies to identify in detail the inclusion and exclusion criteria and to be able to filter out the relevant documents for this work. Therefore, out of 150 documents found and reviewed, we are left with only 20. Figure 1 shows a diagram of the identification and selection carried out.

Data extraction and synthesis: Of 20 articles selected in the previous step, we classified them according to the topic they addressed in Table 2. Subsequently, they were categorized according to the library where they were found (see Table 3) to have details of which library contains the largest number of documents belonging to this work domain.

Phase 3. Document the Review

The most relevant findings in this literature review were in three tables: the first (Table 4) contains information related to SRs in the health and nutrition domain, the second (Table 5) contains information on the main artificial intelligence techniques, methods, and algorithms implemented in SRs, and the last (Table 6), it addresses everything related to the main digital nudging techniques used in FRS.

3 Summary of the Results

Recommendation systems use a variety of techniques to suggest products, services, or information that users are likely to be interested in. The main techniques implemented in these systems are collaborative filtering, content-based filtering, and hybrid models. In this regard, Table 4 shows an overview of these techniques used in recommendation systems in the field of health, where hybrid models are proving to be more useful as they take advantage of both explicit data (ratings, clicks) and implicit data (browsing history, time of use).

It is very important to know the algorithms, methods and technological tools of AI that are used in this field, as they allow us to understand the impact of the implementation of these technologies within recommendation systems and to be able to identify areas for improvement, evaluate the quality of recommendations, among other things. Table 5 shows an overview of the findings found on the AI techniques most used in the development of recommendation systems in the field of healthy eating.

According to the reviewer's work, the nearest neighbor technique (it's great for quick recommendations), matrix factorization (it handles large volumes of data and discovers patterns between user-item interactions), and neural networks (due to their ability to capture complexities and continuously learn to provide more accurate recommendations), are showing good results.

Knowing digital push techniques allows users to be supported in decision-making, evaluate possible biases and, in addition, improve recommendations, aligning them with health and well-being needs. Table 6 shows the most recent information on the types of common nudges used in the design of healthy FRS, with personalized options being much more effective as they significantly improve the effectiveness of the recommendation, adapting to the individual preferences of users, allowing them to improve their experience since recommendations can be more intuitive and less intrusive.

4 Discussion

This review, which systematically identified and analyzed 20 articles published between 2018 and 2024, adds to the existing research literature on HRS in several ways. First, it provides an overview of the recent research landscape on the main techniques of personalized health recommendation in HR.

Second, compared to previous review studies on HRS, this study is more comprehensive and rigorous in terms of literature search, study selection, and analysis of key HRS constructs. We focused on looking for jobs that implement AI tools, and, above all, use digital nudges, since these allow guiding and motivating users towards much healthier choices, improving their experience and satisfaction when interacting with the User Interfaces, although their effects were short-term.

They have been studied and used by various authors, for example, by (15, 16, 17, 18,19), according to [20] the long-term effects of digital nudges are incredibly unknown, which is a major gap that needs to be addressed in future studies because there is no standardization of nudges yet, making it difficult to compare multiple studies.

5 Conclusions

Healthy FRSs are currently having a great boom, as they provide personalized and useful information according to the specific needs and preferences of users, the findings of this review are quite useful to show which Artificial Intelligence technologies are currently used to provide highly personalized and accurate recommendations, considering the nutritional needs of people and the dietary restrictions. Highlighting convolutional neural networks, graphic neural networks and Machine Learning techniques.

Digital nudges, on the other hand, play a very important role, as they can remind users of the importance of choosing nutritious options, encourage moderation in the consumption of unhealthy foods and serve as tools to re-educate the population and improve their quality of life.

Among the main findings, it was found that one of the most widely used digital nudge techniques in recommendation systems in the nutrition domain is the Decision Structure Technique, where cognitive and personalized digital nudges are very useful, and that multiple traffic light tags have been shown to be effective in supporting offline choices.

Today, digital nudges can be found in a variety of recommendation systems, to offer relevant and timely suggestions. This research will support software developers to design healthy FRS to improve the user experience, personalizing content to their specific needs, and thus contribute to the improvement of eating habits.

The key is to use a combination of visual, informational, and motivational techniques to make healthy choices more appealing and accessible. By doing so, these systems can help promote better eating habits and improve the overall health of users.

Acknowledgments. This article was supported by the project Pedagogy of the food-agency-srl-ai program based on digital technology applied in Colombia (FOOD-AGENCY-SRL-AI).

Referencias

1. Yera, R., Alzahrani, A. A., Martínez, L., Rodríguez, R. M.: A systematic review on food recommender systems for diabetic patients. International Journal of Environmental Research and Public Health, vol. 20, no. 5, pp. 4248 (2023) doi: 10.3390/ijerph20054248
2. Ganju, A., Satyan, S., Tanna, V., Menezes, S. R.: AI for improving children's health: a community case study. Frontiers in Artificial Intelligence, vol. 3, pp. 544972 (2021) doi: 10.3389/frai.2020.544972
3. Abhari, S., Lankarani, K. B., Azadbakht, L., Kalhori, S. R. N., Safdari, R., Sefiddashti, S. E., Moradi, S.: Designing and evaluating a nutrition recommender system for improving food security in a developing country. Archives of Iranian Medicine, vol. 26, no. 11, pp. 629-641 (2023) doi: 10.34172/aim.2023.93
4. Chen, Y., Guo, Y., Fan, Q., Zhang, Q., Dong, Y.: Health-aware food recommendation based on knowledge graph and multi-task learning. Foods, vol. 12, no. 10, p. 2079 (2023) doi: 10.3390/foods12102079
5. Forouzandeh, S., Rostami, M., Berahmand, K., Sheikhpour, R.: Health-aware food recommendation system with dual attention in heterogeneous graphs. Computers in Biology and Medicine, vol. 169, pp. 107882 (2024) doi: 10.1016/j.compbiomed.2023.107882
6. Karlsen, R., Andersen, A.: Recommendations with a nudge. Technologies, vol. 7, no. 2, p. 45 (2019) doi: 10.3390/technologies7020045
7. Dolgopolova, I., Toscano, A., Roosen, J.: Different shades of nudges: Moderating effects of individual characteristics and states on the effectiveness of nudges during a fast-food order. Sustainability, vol. 13, no. 23, p. 13347 (2021) doi: 10.3390/su132313347
8. Starke, A., El-Majjodi, A., Trattner, C.: Boosting health? examining the role of nutrition labels and preference elicitation methods in food recommendation. Interfaces and Human Decision Making for Recommender Systems, vol. 2022, pp. 67–84 (2022)
9. El-Majjodi, A., Starke, A. D., Trattner, C.: Nudging towards health? examining the merits of nutrition labels and personalization in a recipe recommender system. In: Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization, pp. 48–56 (2022) doi: 10.1145/3503252.3531312
10. Yi, S., Kanetkar, V., Brauer, P.: Nudging food service users to choose fruit-and vegetable-rich items: Five field studies. Appetite, vol. 173, p. 105978 (2022) doi: 10.1016/j.appet.2022.105978
11. Castiglia, G., El-Majjodi, A., Starke, A. D., Narducci, F., Deldjoo, Y., Caló, F.: Nudging towards health in a conversational food recommender system using multi-modal interactions and nutrition labels. In: Fourth Knowledge-aware and Conversational Recommender Systems Workshop (KaRS), pp. 18–23 (2022)
12. Ytreberg, N. S., Alfnes, F., van-Oort, B.: Mapping of the digital climate nudges in Nordic online grocery stores. Sustainable Production and Consumption, vol. 37, pp. 202–212 (2023) doi: 10.1016/j.spc.2023.02.018

13. El-Majjodi, A., Starke, A. D., Elahi, M., Trattner, C.: The Interplay between food knowledge, nudges, and preference elicitation methods determines the evaluation of a recipe recommender system. In: IntRS@ RecSys, pp. 1–18 (2023)
14. Chiam, J., Lim, A., Nott, C., Mark, N., Teredesai, A., Shinde, S.: Co-pilot for health: personalized algorithmic AI nudging to improve health outcomes. (2024) arXiv:2401.10816.
15. De-Bauw, M., De-la-Revilla, L. S., Poppe, V., Matthys, C., Vranken, L.: Digital nudges to stimulate healthy and pro-environmental food choices in E-groceries. Appetite, vol. 172, p. 105971 (2022) doi: 10.1016/j.appet.2022.105971
16. Gatautis, R., Vitkauskaite, E., Gadeikiene, A., Piligrimiene, Z.: Gamification as a mean of driving online consumer behaviour: SOR model perspective. Engineering Economics, vol. 27, no. 1, pp. 90–97 (2016) doi: 10.5755/j01.ee.27.1.13198
17. Mills, S.: Finding the 'nudge'in hypernudge. Technology in Society, vol. 71, p. 102117 (2022) doi: 10.1016/j.techsoc.2022.102117
18. Nori, R., Zucchelli, M. M., Giancola, M., Palmiero, M., Verde, P., Giannini, A. M., Piccardi, L.: GPS digital nudge to limit road crashes in non-expert drivers. Behavioral Sciences, vol. 12, no. 6, p. 165 (2022) doi: 10.3390/bs12060165
19. Zimmermann, V., Renaud, K.: The nudge puzzle: matching nudge interventions to cybersecurity decisions. ACM Transactions on Computer-Human Interaction (TOCHI), vol. 28, no. 1, pp. 1–45 (2021) doi: 10.1145/34298
20. Gyulai, Z., Revesz, B.: Nudging in the digital world: An up-to-date systematic literature review. In: Proceedings of the European Marketing Academy, pp. 117–268 (2023)
21. Abhari, S., Safdari, R., Azadbakht, L., Lankarani, K. B., Kalhori, S. R. N., Honarvar, B., Jalilpiran, Y.: A systematic review of nutrition recommendation systems: with focus on technical aspects. Journal of biomedical physics & engineering, vol. 9, no. 6, pp. 591–602 (2019) doi: 10.31661/jbpe.v0i0.1248
22. Nouh, R. M., Lee, H. H., Lee, W. J., Lee, J. D.: A smart recommender based on hybrid learning methods for personal well-being services. Sensors, vol. 19, no. 2, p. 431 (2019) doi: 10.3390/s19020431
23. Cai, Y., Yu, F., Kumar, M., Gladney, R., Mostafa, J.: Health recommender systems development, usage, and evaluation from 2010 to 2022: A scoping review. International Journal of Environmental Research and Public Health, vol. 19, no. 22, p. 15115 (2022) doi: 0.3390/ijerph192215115
24. Sun, Y., Zhou, J., Ji, M., Pei, L., Wang, Z.: Development and evaluation of health recommender systems: systematic scoping review and evidence mapping. Journal of Medical Internet Research, vol. 25 (2023)
25. Chavan, P., Thoms, B., Isaacs, J.: A recommender system for healthy food choices: building a hybrid model for recipe recommendations using big data sets (2021)
26. Chow, Y. Y., Haw, S. C., Naveen, P., Anaam, E. A., Mahdin, H. B.: Food recommender system: a review on techniques, datasets and evaluation metrics. Journal of System and Management Sciences, vol. 13, no. 5, pp. 153–168 (2023) doi: 10.33168/JSMS.2023.0510

ISSN 1870-4069

Classification of Zika and Dengue Clinical Data Using Feature Encoding and Machine Learning Techniques

Elí Cruz-Parada[1], Guillermina Vivar-Estudillo[2],
Eduardo Pérez-Campos[1], Carlos Lastre-Domínguez[1]

[1] Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca,
División de Posgrado e Investigación,
Mexico

[2] Universidad Autónoma “Benito Juárez” de Oaxaca,
Facultad de Sistemas Biológicos e Innovación Tecnológica,
Mexico

`Eli.cruz.parada@gmail.com,gvivar.cat@uabjo.mx,`
`{pcampos,carlos.lastre}@itoaxaca.edu.mx`

Abstract. Dengue and Zika are viral diseases primarily transmitted to humans through bites from infected female Aedes aegypti mosquitoes, posing considerable medical concerns due to their potential severity. These illnesses can lead to fatal hemorrhagic fevers and have affected around 300 million people globally. Early diagnosis is crucial for effective treatment. Despite extensive research over recent decades, achieving diagnostic accuracy remains challenging. This study introduces a novel method for organizing clinical data to enhance the identification of Zika and Dengue by utilizing symptoms as features extracted from clinical studies and applying machine learning techniques for classification tasks. Rigorous statistical analysis using ANOVA and Kruskal-Wallis tests revealed p-values below 0.05, indicating significant findings. Additionally, the classifiers examined demonstrated AUCs and F1 scores exceeding 96%, highlighting their effectiveness. This approach aims to improve diagnostic precision, thereby facilitating timely intervention and reducing the impact of these diseases on global health.

Keywords: Tropical diseases, dengue, sika, clinical data, machine learning.

1 Introduction

Dengue and Zika fever are viral diseases transmitted to humans primarily through the bite of an infected female Aedes aegypti mosquito. While other species of the Aedes genus can also transmit these viruses, their role is generally secondary [1]. Once a mosquito becomes a vector, it remains so for its entire lifespan. Both diseases are more prevalent in tropical and subtropical regions, with an increasing incidence observed in recent years. The World Health Organization (WHO) reported a dramatic rise in dengue cases, from 505,430 cases in 2000 to 5.2 million in 2019 [2].

Dengue is often asymptomatic; however, when symptoms do appear, they typically resolve within one to two weeks. Despite being classified as mild or moderate, dengue can develop into a severe form, involving bleeding and requiring hospitalization due to the risk of fatality. Common symptoms include fever, headache, retro-orbital pain, nausea, and vomiting [3]. A second dengue infection often leads to more severe illness, which can be misdiagnosed as other febrile illnesses. Similarly, Zika fever, caused by the ZIKV virus, is transmitted by the same mosquito species and can also be spread through sexual transmission.

Symptoms include rash, itching, non-purulent conjunctivitis, arthralgia, myalgia, and fever. Only about one in four infected individuals exhibit symptoms, which are generally mild and last for 2 to 7 days. The clinical presentation is often similar to dengue or Chikungunya, necessitating laboratory confirmation [4]. Currently, there is no specific treatment for dengue, with management focusing on pain relief, while avoiding nonsteroidal anti-inflammatory drugs (NSAIDs) due to bleeding risks.

Research teams worldwide are increasingly using machine learning and data mining techniques to improve disease diagnosis. For example, decision trees have been successfully used to differentiate between tropical infections [5]. In Paraguay, researchers achieved an average accuracy of 96% using Support Vector Machine classifiers and Artificial Neural Networks [6]. An Android application named GZC-DIAG outperformed resident physicians in diagnosing diseases with a 96.88% success rate [7].

Innovations continue with machine learning integrated into laboratory tests, such as peripheral blood smear (PBS) analysis, showing promising results with up to 95.74% accuracy in detecting Dengue Virus (DENV) [8]. Despite challenges related to data scarcity in specific clinical analyses, these advancements underscore the potential of AI in transforming diagnostic capabilities.

Moreover, beyond diagnosis, efforts are being made to predict the risks associated with diseases like dengue. Studies have demonstrated accuracies ranging from 70% to 96.27% using various techniques, including bioelectrical impedance analysis (BIA) and neural networks [9, 10]. These predictive models not only aid in diagnosing but also in assessing the prognosis and potential complications of patients.

Looking forward, the development of more affordable and portable diagnostic tools, such as biosensor devices, represents a significant advancement, particularly for resource-limited settings. Furthermore, AI-driven models have proven effective in forecasting disease outbreaks, achieving up to 89.25% accuracy in predicting dengue outbreaks [11]. In India, AI has been used to predict outbreaks and diagnose diseases like Zika using data from users and environmental factors. The processing time is 0.15 s with 91.25% accuracy [12].

The diagnosis by 3D super-resolution microscopy images has been used in Zika, these images are taken from the endoplasmic reticulum (ER). Deep learning techniques were able to identify morphological changes in the ER caused by the virus [13]. Similarly, ensemble methods were applied to identify cases of congenital Zika, these were based on the U.S. Zika Pregnancy and Infant Registry (USZPIR) and the Zika Active Pregnancy Surveillance System (ZAPSS) of Puerto Rico, and although it presented a high sensitivity (96% for USZPIR and 97% for ZAPSS), the model was specifically designed for this dataset [14].

Research on using Electrocardiogram (ECG)-derived heart rate variability (HRV) metrics and machine learning (ML) models to predict infant exposure to Zika virus (ZIKV) has been conducted. In a study of 21 infants with an average age of 15 months, a cubic support vector machine classifier was applied to their ECGs [15]. The research

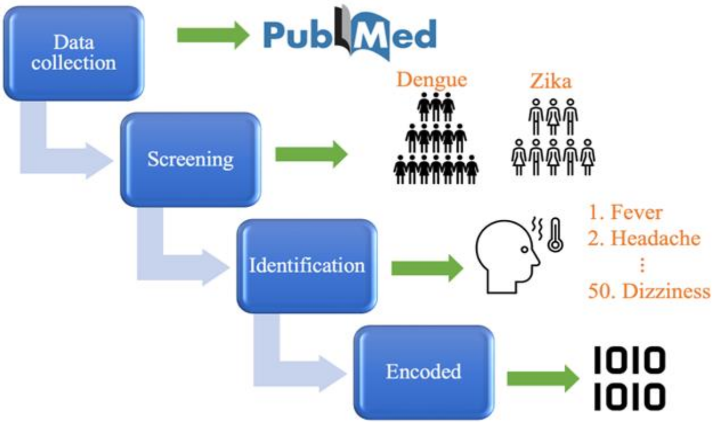


Fig.1. Processing system for a codified symptoms database.

Table 1. Codification of symptoms.

# Patient	Pathology	Fever	Headache	Myalgia	Nausea	Rash
1	Zika	0	0	0	0	1
2	Dengue	1	1	0	0	0
3	Dengue	1	1	1	0	0
4	Zika	0	1	0	0	1

team reported that their model was able to differentiate between infants affected by Zika, non-affected by Zika, and those not affected by the virus with a predictive value of 86%. However, there is some dispute about whether HRV is a specific attribute of Zika.

A review, reported in 2022, concluded that machine learning and deep learning techniques for diagnosing arboviral diseases focus mainly on dengue and do not effectively differentiate between more than two different pathologies. It was also noted that decision tree-based techniques are the most used [16].

The approaches mentioned above yield important findings, but they are not sufficient for accurately diagnosing tropical diseases. We propose a feature coding method that takes into account the 50 most common symptoms and utilizes advanced machine-learning techniques to identify Zika and Dengue pathology. This work comprises several sections. Section 2 provides a brief overview of the database. Section 3 offers an in-depth analysis of features using ANOVA and Kruskal Wallis, including p-value analysis. Section 4 outlines the performance of classifiers in determining Zika and Dengue. Section 5 discusses the results of this work. Finally, in Section 6, we present our conclusions.

2 Database

As depicted in Fig. 1, we have developed a methodology for establishing a database consisting of four subprocesses: data collection, screening, identification, and encoding. We will now elaborate on each stage.

2.1 Data Collection and Screening

To gather reliable data and establish a database, a search was conducted in the PubMed database using the keywords "dengue," "prevalence," and "clinical symptoms." The selected studies had to meet specific criteria. Firstly, they needed to report on at least 50 patients diagnosed with dengue through laboratory tests. Secondly, they had to mention at least five symptoms observed in confirmed dengue patients. Thirdly, they were required to provide the number of patients affected by each symptom, including the mortality rate. Finally, they needed to confirm that the symptoms were observed within the first four days of the disease.

To gather Zika data, a search was conducted in the PubMed database using the keywords "Zika fever," "prevalence," and "clinical symptoms." However, as there were significantly more papers for Dengue (1,999 papers) than for Zika (673 papers), the selection criteria were adjusted as follows:

1. The study must report on at least 20 patients diagnosed with Zika through laboratory tests.
2. The study should mention more than five different symptoms observed in confirmed Zika patients.
3. The study should provide the number of patients who experienced each symptom.
4. The study should present the mortality rate.
5. The symptoms should have been observed within the first four days of the disease.

The following criteria are used to select papers: Each paper is from relevant journals, such as *PLOS Neglected Tropical Diseases* [17], *BMC Infectious Diseases* [18], *Annals of Medicine* [19], *The Lancet Infectious Diseases* [20], among others. We aim to minimize redundant investigations to ensure that each study provides unique and valuable information. To avoid bias, data is selected from different hospitals, years, and countries for each study.

2.2 Identification

For the identification processing, the age, gender, economic status, and nationality of the patients were not recorded in any of the cases. After collecting the necessary data, 22,379 dengue-confirmed patients and 7,135 Zika-confirmed patients were observed for up to 37 and 34 different symptoms, respectively. In total, 20 common and 30 other symptoms were observed, making up 50 symptoms.

2.3 Encoded

The patients' symptoms were coded, a label was added to each symptom indicating “1” if the symptom was present and “0” if it was absent. To illustrate the codification, table 1 considers four patients and five symptoms: Patient 1 is coded as 00001 with Zika pathology. Subsequently, we established a database coded for Zika and Dengue symptoms. Next, a statistical analysis is performed to find significance.

Table 2. F-value and H-value calculated for some symptoms.

Symptoms	**ANOVA**	**Kruskal-Wallis**
Rash	1.6871×10^4	1.0735×10^4
Fever	1.4924×10^4	9.9118×10^3
Conjunctivitis	6.4686×10^3	5.3059×10^3
Pruritus	3.1877×10^3	2.8770×10^3
Low back pain	2.3153×10^3	2.1469×10^3
Retro orbital pain	1.6269×10^3	1.5419×10^3
Sore throat	1.1335×10^3	1.0916×10^3
Edema	1.1249×10^3	1.0836×10^3

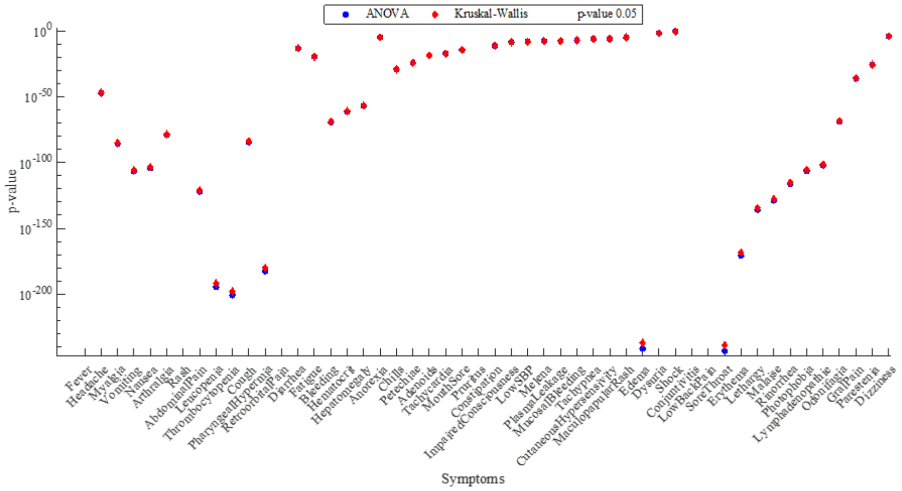


Fig.2. p-value with ANOVA and Kruskal-Wallis analysis.

3 Features Statistical Analysis

3.1 ANOVA

The features are evaluated through the ANOVA test, calculating the F-value by mean equation (1):

$$F = \frac{MSB}{MSW}. \tag{1}$$

To calculate the mean square between (MSB) groups for the ANOVA test, it should start by calculating the mean of each group.

Then, calculate the overall mean (the mean of all data points combined), and finally calculate the sum of squares between groups (SSB) by equation (2).

$$SSB = \sum_{i=1}^{k} n_i(\bar{X}_i - \bar{X})^2, \tag{2}$$

where n_i is the number of observations in group i, $\bar{X}_i$ is the mean of group i and $\bar{X}$ is the overall mean. The degrees of freedom between groups (df_b) can be computed by equation (3):

$$df = k - 1, \tag{3}$$

where k is the number of groups. Calculated the SSB and df_b, the MSB is computed by equation (4):

$$MSB = \frac{SSB}{df_b}. \tag{4}$$

To compute mean square within (MSW) groups for the ANOVA test, initially, the sum of squares within groups (SSW) is calculated by equation (5):

$$SSW = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2, \tag{5}$$

where X_{ij} is the j-th observation in group i. The degrees of freedom within (df_w) are calculated by equation (6):

$$df_w = N - 1, \tag{6}$$

where N is the total number of observations across all groups. Finally, the MSW is calculated by the equation (7):

$$MSW = \frac{SSW}{df_w}. \tag{7}$$

3.2 Kruskal-Wallis Test

A Kruskal-Wallis analysis was also conducted using the statistical test (H or χ^2 chi-square) represented by equation (11):

$$H = \frac{12}{N_s\,(N_s + 1)} \sum_{i=1}^{k} \left(\frac{SS_i}{n_i}\right) - 3(N_s + 1), \tag{8}$$

where k is the number of groups, N_s is the total number of observations, n_i is the number of observations in the i-th group and SS_i is the sum of the squared ranks within the i-th group.

In Table 2, ANOVA and Kruskal-Wallis analyses are being performed to compare symptoms such as rash, fever, conjunctivitis, pruritus, low back pain, retro-orbital pain, sore throat, and edema. As can see in Fig. 2, the statistical analysis provides a detailed p-value analysis to determine the statistical significance of these symptoms, with the majority showing a p-value of less than 0.05.

4 Classification

We are evaluating five classifiers: cubic SVM, quadratic SVM, Gaussian, fine KNN, and weighted KNN. We conducted an AUC analysis to assess the overall discriminative ability of the classifiers between the positive and negative classes. The AUC is determined by the true positive rate (TPR) and the false positive rate (FPR), which are calculated by equation (9) and equation (10), respectively:

$$TPR = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + Falses\ Positives\ (FP)}, \tag{9}$$

$$FPR = \frac{False\ Positives\ (FP)}{False\ Positives\ (FP) + True\ Negatives\ (TN)}. \tag{10}$$

As seen in Fig. 3, the classifiers achieve an AUC near 0.96, indicating a high level of performance.

The F1-score was calculated, and the accuracy of the test was measured using equation (11):

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{11}$$

where precision and recall are calculated by equation (12) and (13), respectively:

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FN)}, \tag{12}$$

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)}. \tag{13}$$

Under 100 iterations and a holdout of 80/20, we performed calculations to determine the average Area Under the Curve (AvgAUC) and its standard deviation (StdAUC), as well as the average F1-Score (AvgF1) and its standard deviation (StdF1) for balanced sets, with the minority class being the Zika pathology. Subsampling was conducted for the majority class, resulting in three sets.

The first and second sets consisted of 7135 data for Zika and Dengue, respectively, while the third set comprised 8109 data for Dengue and 7135 for Zika, with the majority class being dengue. Tables 3, 4, and 5 display the performance of different classifiers, all of which exceeded the 96% threshold.

5 Discussion

The effectiveness of machine learning techniques also depends on how the data is represented. Therefore, it is essential to conduct a statistical analysis to distinguish between Zika and Dengue based on their symptoms. After performing ANOVA and Kruskal-Wallis tests, it was determined that symptoms such as fever, rash, conjunctivitis, pruritus, low back pain, retro-orbital pain, sore throat, and edema exhibited the most statistical significance.

Table 3. AUC Average performance for the first set.

Classifier	**AvgAUC**	**StdAUC**	**AvgF1**	**StdF1**
Fine KNN	0.9689	0.00285	0.9683	0.0029503
Cubic SVM	0.9862	0.00282	0.9817	0.0019973
Medium Gaussian SVM	0.9870	0.00318	0.9824	0.0018754
Quadratic SVM	0.9871	0.00344	0.9821	0.0018718
Weighted SVM	0.9758	0.00266	0.9626	0.0030827

Table 4. AUC Average performance for the second set.

Classifier	**AvgAUC**	**StdAUC**	**AvgF1**	**StdF1**
Fine KNN	0.9703	0.00260	0.9694	0.00276
Cubic SVM	0.9865	0.00299	0.9816	0.00228
Medium Gaussian SVM	0.9890	0.00269	0.9822	0.00219
Quadratic SVM	0.9750	0.00302	0.9823	0.00198
Weighted SVM	0.9758	0.00266	0.9618	0.00303

Table 5. AUC Average performance for the third set.

Classifier	**AvgAUC**	**StdAUC**	**AvgF1**	**StdF1**
Fine KNN	0.96869	0.00330	0.9676	0.00351
Cubic SVM	0.98496	0.00296	0.9813	0.00265
Medium Gaussian SVM	0.98591	0.00290	0.9818	0.00250
Quadratic SVM	0.98801	0.00355	0.9816	0.00248
Weighted SM	0.97419	0.00287	0.9609	0.00378

Therefore, these symptoms are crucial for predicting diseases using machine learning techniques. Despite utilizing 50 symptoms, the computational cost was not significantly affected.

While this did not significantly impact the current research, adding data from additional diseases could substantially increase computational costs.

The study also identified the best-performing classifiers. Decision tree-based classifiers and ensemble models like random forest, adaboost, and gradient boosting exhibited acceptable performance levels of around 90%, although they were found to be less effective than SVM-based models and certain non-parametric algorithms such as k-nearest neighbors (KNN). This study specifically focused on a limited number of diseases, namely Zika and Dengue. If additional diseases are included, it will be necessary to analyze more symptoms and conduct more rigorous statistical analyses.

6 Conclusions

We have developed a new method for organizing clinical data to help detect diseases such as Zika and Dengue. Our analysis revealed significant differences, with a p-value of 0.05, using ANOVA and Kruskal-Wallis tests. When we tested the features using various classifiers, we achieved an average performance of over 96%, ranging from 96% to 99%, through iterative training and testing. The classification was performed using an 80/20 holdout and 100 iterations, and the results from the classifiers demonstrate that the data is well represented.

In our future research, we aim to analyze more clinical data and explore other tropical diseases like Chikungunya. We also plan to use different machine learning models for classification and investigate the incorporation of deep learning techniques.

References

1. Malavige, G. N., Fernando, S., Fernando, D. J., Seneviratne, S. L. : Dengue viral infections. Postgraduate Medical Journal, vol. 80, no. 948, pp. 588–601 (2004) doi: 10.1136/pgmj.2004.019638
2. World Health Organization: Dengue and severe dengue. https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue, last accessed 2023/05/10
3. Guo, C., Zhou, Z., Wen, Z., Liu, Y., Zeng, C., Xiao, D., Ou, M., Han, Y., Huang, S., Liu, D., Ye, X., Zou, X., Wu, J., Wang, H., Zeng, E. Y., Jing, C., Yang, G.: Global epidemiology of dengue outbreaks in 1990-2015: A systematic review and meta-analysis. Frontiers in cellular and infection microbiology, vol. 7, no. 317 (2017) doi: 10.3389/fcimb.2017.00317
4. World Health Organization: Zika virus. https://www.who.int/news-room/fact-sheets/detail/zika-virus, last accessed 2023/05/10
5. Shenoy, S., Rajan, A. K., Rashid, M., Chandran, V. P., Poojari, P. G., Kunhikatta, V., Acharya, D., Nair, S., Varma, M., Thunga, G.: Artificial intelligence in differentiating tropical infections: A step ahead. PLoS neglected tropical diseases, vol. 16, no. 6 (2022)
6. Mello-Román, J. D., Mello-Román, J. C., Gómez-Guerrero, S., García-Torres, M.: Predictive models for the medical diagnosis of dengue: A case study in Paraguay. Computational and Mathematical Methods in Medicine (2019) doi: 10.1155/2019/7307803
7. Rodríguez-De-Araújo, A. P., Macedo-de-Araujo, M. C., Coutinho-Cavalcanti, T., de-Lacerda-Vidal, C. F., Gomes-Netto-Monte-da-Silva, M.: DZC DIAG: mobile application based on expert system to aid in the diagnosis of dengue, zika, and chikungunya. Medical & Biological Engineering & Computing, vol. 58, no. 11, pp. 2657–2672 (2020) doi: 10.1007/s11517-020-02233-6
8. Mayrose, H., Bairy, G. M., Sampathila, N., Belurkar, S., Saravu, K.: Machine learning-based detection of dengue from blood smear images utilizing platelet and lymphocyte characteristics. Diagnostics (Basel, Switzerland), vol. 13, no. 2, pp. 220 (2023) doi: 10.3390/diagnostics13020220
9. Faisal, T., Ibrahim, F., Taib, M. N.: A noninvasive intelligent approach for predicting the risk in dengue patients. Expert Systems with Applications, vol. 37, no. 3, pp. 2175–2181 (2010) doi: 10.1016/j.eswa.2009.07.060
10. Ibrahim, F., Faisal, T., Salim, M. I., Taib, M. N.: Non-invasive diagnosis of risk in dengue patients using bioelectrical impedance analysis and artificial neural network. Medical & biological engineering & computing, vol. 48, no. 11, pp. 1141–1148 (2010) doi: 10.1007/s11517-010-0669-z

11. Anggraini-Ningrum, D. N., Yu-Chuan, J. L., Hsu, C. Y., Solihuddin-Muhtar, M., Pandu-Suhito, H. P.: Artificial intelligence approach for severe dengue early warning system. Stud Health Technol Inform, pp. 881–885 (2023) doi:10.3233/SHTI231091
12. Dadheech, P., Mehbodniya, A., Tiwari, S., Kumar, S., Singh, P., Gupta, S., Atiglah, H. K.: Zika virus prediction using AI-driven technology and hybrid optimization algorithm in healthcare. Journal of Healthcare Engineering, vol. 12, pp. 2793850 (2022) doi: 10.1155/ 2022/2793850
13. Long, R. K., Moriarty, K. P., Cardoen, B., Gao, G., Vogl, A. W., Jean, F., Nabi, I. R.: Super resolution microscopy and deep learning identify zika virus reorganization of the endoplasmic reticulum. Scientific Reports, vol. 10, no. 1, p. 20937 (2020) doi: 10.1038/ s41598-020-77170-3
14. Lusk, R., Zimmerman, J., van-Maldeghem, K., Kim, S., Roth, N. M., Lavinder, J., Fulton, A., Raycraft, M., Ellington, S. R., Galang, R. R.: Exploratory analysis of machine learning approaches for surveillance of zika-associated birth defects. Birth Defects Research, vol. 112, no. 18, pp. 1450–1460 (2020) doi: 10.1002/bdr2.1767
15. Herry, C. L., Soares, H. M. F., Schuler-Faccini, L., Frasch, M. G.: Machine learning model on heart rate variability metrics identifies asymptomatic toddlers exposed to zika virus during pregnancy. Physiological Measurement, vol. 42, no. 5 (2021) doi: 10.1088/1361-6579/ac010e
16. da Silva-Neto, S. R., Tabosa-Oliveira, T., Teixeira, I. V., Aguiar de Oliveira, S. B., Souza-Sampaio, V., Lynn, T., Endo, P. T.: Machine learning and deep learning techniques to support clinical diagnosis of arboviral diseases: A systematic review. PLoS Neglected Tropical Diseases, vol. 16, no. 1 (2022) doi: 10.1371/journal.pntd.0010061
17. Rafi, A., Nahar-Mousumi, A., Ahmed, R., Haque-Chowdhury, R., Wadood, A., Hossain, G.: Dengue epidemic in a nonendemic zone of Bangladesh: Clinical and laboratory profiles of patients. PLOS Neglected Tropical Diseases, vol. 14, no. 10 (2020) doi: 10.1371/journal. pntd.0008567
18. Hasan, M. J., Tabassum, T., Sharif, M., Khan, M. A. S., Bipasha, A. R., Basher, A., Amin, M. R.: Comparison of clinical manifestation of dengue fever in Bangladesh: an observation over a decade. BMC infectious diseases, vol. 21, no. 1, pp. 1113 (2021) doi; 10.1186/s1287 9-021-06788-z
19. Asaga-Mac, P., Tadele, M., Airiohuodion, P. E., Nisansala, T., Zubair, S., Aigohbahi, J., Panning, M.: Dengue and zika seropositivity, burden, endemicity, and cocirculation antibodies in Nigeria. Annals of Medicine, vol. 55, no. 1, pp. 652–662 (2023) doi: 10.1080/ 07853890.2023.2175903
20. Marc-Ho, Z. J., Chanditha-Hapuarachchi, H., Barkham, T., Chow, A., Ng, L. C., Vernon-Lee, J. M., Sin-Leo, Y., Prem, K., Georgina-Lim, Y. H., de-Sessions, P. F., Rabaa, M. A., Seng-Chong, C., Hua-Tan, C., Rajarethinam, J., Tan, J., Anderson, D. E., Ong, X., Cook, A. R., Chong, C. Y., Hsu, L. Y., et. al.: Outbreak of zika virus infection in Singapore: an epidemiological, entomological, virological, and clinical analysis. The Lancet Infectious Diseases, vol. 17, no. 8, pp. 813–821 (2017)

ISSN 1870-4069

Transfer Learning: Shallow and Deep Neural Models

Diego Uribe, Enrique Cuan, Elisa Urquizo

Tecnológico Nacional de México,
Instituto Tecnológico de La Laguna,
Departamento de Sistemas y Computación,
Mexico

{duribea, ecuand, eurquizob}@lalaguna.tecnm.mx

Abstract. Since developing general-purpose large language models to handle a variety of tasks can be expensive for many businesses, transfer learning plays a crucial role in many natural language tasks nowadays. The pre-trained and fine-tuning framework for transferring and tuning the generic knowledge produced by a large language model (LLM) is a common ground for many natural language tasks. In this work, we focus our attention on how much tuning of the semantic representations (i.e. generic knowledge) obtained from BERT is required to perform downstream language tasks. We analyze the impact of some deep learning models, such as recurrent neural networks, on the fine-tuning process for a downstream classification task. To consider similarities and differences between the deep learning models and the corresponding optimization of the classification task, a dataset of four different categories of short answer responses is used in our empirical experimentation. In this way, we enrich the comparison of the tuning required to optimize the semantic representations obtained from a pre-trained BERT model.

Keywords: Transfer learning, pre-trained language model, BERT.

1 Introduction

In this paper we explore whether there is a meaningful contrast in the use of different deep learning models when transferring the generic knowledge produced by a pre-trained language model. In the context of natural language processing, learning textual representations, the generic knowledge to be transferred, plays a crucial role in multiple language tasks like question answering or classification or natural language generation. These textual representations are learned from unsupervised learning methods and transferred to a supervised learning task. Said in another way, transfer learning occurs when generic representations have been learned to make a subsequent learning task easier [5]. And it is precisely that the pre-trained and fine-tuned framework has emerged as a powerful technique to facilitate transfer learning. Building a pre-trained language model relies on self-supervised learning (SSL), a type of unsupervised learning.

By making use of large amounts of unlabeled text data, SSL learns generic representations useful across many linguistic tasks [2]. The most crucial aspect in the development of a pre-trained model is the definition of the unsupervised learning tasks such as masked language modeling (MLM) [18], next sentence prediction (NSP) [12], sentence order prediction (SOP) [9], etc. These pre-training tasks make the difference between the large number of pre-trained models available to be used across a range of downstream language tasks [22]. In this work we focus our attention on the bidirectional transformer encoder known as BERT: Bidirectional Encoder Representations from Transformers [3], a pre-trained language model based on a bidirectional transformer encoder which is characterized by a bidirectional self-attention mechanism to produce contextual embeddings. More precisely, among the wide range of available variants of BERT, in this work we make use of the compact BERT model [20, 19].

Since a large language model such as the original BERT has a high computational cost, the compact BERT model was created with the purpose of not only reducing the computational cost but also using the same self-supervised learning paradigm in its development. Indeed, building a compact model proved to be possible by applying the standard pre-training and fine-tuning process but a different training strategy, based on a compression technique known as knowledge distillation, was implemented.

Briefly, this distillation technique consists of a student-teacher training method where the teacher transfers knowledge to the student through its predictions for unlabeled training examples. A deep description is given in section 3. As we initially said, we explore in this work whether there is a meaningful contrast in the use of different deep learning models when taking the generic knowledge produced by a pre-trained language model. In other words, we explore the impact of taking and further tuning the linguistic representations obtained from the compact BERT model via deep learning models such as simple recurrent networks (SimpleRNN), long short term memory networks (LSTM) and Bi-directional networks.

In fact, we perform a downstream task as classification by implementing a classifier learning model with each variant of recurrent neural networks for tuning the obtained representations to the peculiarities of a downstream task as short answer responses classification. The collection of short answer responses was created with the intention of automated assessment of written responses. Each instance in the collection denotes a short answer corresponding to a particular story of a specific domain where the range of the score is three: 0, 1, or 2. In other words, the fine-tuning process performs a downstream task as multi-class classification where a short answer is assigned into one of the multiple rubrics of the responses. Thus, the primary contributions of our work are summarized as follows:

- Our work provides insights about the impact of deep transfer learning with recurrent neural networks. The transfer learning for each variant of recurrent neural networks is described to consider similarities and differences between them.
- We conduct an empirical evaluation on the use of recurrent neural networks for transfer learning. The fine tuning process is implemented on a downstream classification task with a deep learning model defined in terms of the semantic representations produced by the compact BERT model.
- To enrich the experimentation, we try two different configurations for each variant of RNNs.

2 Related Work

In this section we briefly describe some interesting works about transfer learning and pre-trained language models. Developing a pre-trained language model to produce general-purpose knowledge leads to developing modern techniques for transfer learning in NLP. Azunre has written a fully comprehensive guide about transfer learning techniques for the customization of pre-trained language models.

How to use transfer learning to reproduce state-of-the-art results for downstream tasks, especially when limited resources, such as the availability of label data, are a common scenario in many language understanding tasks [1]. Another important work about the relevance of transfer learning is displayed by Raffel et al. [14]. Given that transferring general knowledge produced by a pre-trained language model is nowadays a common practice in many language tasks, transfer learning emerges as an important research topic. The purpose of this work is to propose a text-to-text framework for the systematic study of the multiple affairs that transfer learning entails:

- Pre-training objectives,
- Unlabeled datasets,
- Benchmarks,
- Fine-tuning methods.

In other words, this work proposes a unified approach to compare the effectiveness of various transfer learning objectives and fine-tuning methods, as well the use of unlabeled datasets. In summary, the purpose is to provide insights about transfer learning from a systematic framework to determine where the field stands.

An excellent paper about the impact of pretrained language models in NLP is the work developed by Qiu et al. [13]. The major contribution of this work is a comprehensive and exhaustive review of pretrained models for NLP. For a better description of the pretrained models, the authors built a taxonomy which categorize existing pretrained models from four different perspectives:

- **Representation Type**: Contextual and non-contextual models for downstream tasks.
- **Architectures**: The network structure and its components such as Transformer encoder and decoder.
- **Task Types**: Type of pre-training tasks.
- **Extensions**: Design of pretrained models for diverse scenarios.

The problem of adapting the general language knowledge to downstream tasks is also contemplated from the perspective of transfer learning and fine-tuning strategies. There have been numerous works on improving BERT such as RoBERTa [11]. This encoder is more robust than BERT, and is trained using much more training data.

ALBERT [9] is another work that lowers the memory consumption and increases the training speed of BERT. These variants have also been fine-tuned for various NLP tasks. Lin et al. used self-attention to extract interpretable sentence embeddings [10]. They use a 2-D matrix to represent the embedding, with each row of the matrix attending on a different part of the sentence. In this way, different aspects of the sentence are extracted into multiple-vector representation. Experimental results over 3 different tasks show that the model outperforms other sentence embedding models by a significant margin.

3 BERT: The Pre-trained Language Model

Here we explain the pre-trained language model used in our research work, a variant of BERT: the compact BERT model. In fact, a fair description of knowledge distillation, the pre-training task used in the development of this small language model, is given in this section. But we first briefly take a look at BERT and its self-attention mechanism that has impacted the world of NLP.

3.1 BERT: Bidirectional Encoder Representations from Transformers

In its broadest sense, the transformer consists of an encoder-decoder architecture. However, BERT is a transformer model that includes only the encoder component. Unlike other popular embedding models (e.g. word2vec) that produce static embeddings irrespective of the context, BERT generates dynamic embeddings based on the context so multiple embeddings are produced for the multiple contexts in which a particular word can be used [3]. In order to generate context-based embeddings, the attention mechanism of the transformer plays a crucial role in the encoding process.

Self-attention, a special type of attention, emerged as a more efficient alternative to overcome the limitations of the RNNs: capturing long-term dependencies is one of the major challenges with RNNs [21]. Self-attention takes a holistic approach to the analysis of the linguistic elements: instead of considering only the previous elements in the input, self-attention compares each element with all the sequence elements in order to understand how words relate to each other over long distances. In fact, the output of a particular element y_i depends on the comparisons between the input x_i and the preceding and following elements x_j. A formal description of the output values (vector y) is based on three concepts:

- **Query**: The current focus of attention.
- **Key**: Preceding and following input to be compared with the current focus of attention.
- **Value**: Computation of the output for the current focus of attention.

In this way, each element of the input vector **x** is represented in terms of these concepts and the corresponding weights:

$$q_i = W^{Qx_i}; \qquad k_i = W^{Kx_i}; \qquad v_i = W^{Vx_i}. \tag{1}$$

Then, the output y_i corresponding to each input element x_i is:

$$y_i = \sum_{j=i}^{n} \alpha_{ij} v_j, \tag{2}$$

where the alpha weights represent the proportional relevance of each input to the current focus of attention:

$$\alpha_{ij} = \frac{\exp(\text{score}_{ij})}{\sum_{k=1}^{n} \exp(\text{score}_{ik})}, \tag{3}$$

$$\text{score}_{ij} = q_i k_j. \tag{4}$$

Thus the comparison of each element with the rest of the sequence elements take place in parallel. This means simultaneous access to all sequence elements and therefore simultaneous computation of the relevance of each sequence element. In this way, the step-by-step processing of intermediate recurrent connections is eliminated.

3.2 Compact Model

The basic idea of the **C**ompact BERT model is to start with an initial model trained on MLM (the student) to eventually improve its performance by knowledge distillation from a large language model (the teacher) [20]. This model consists of $L = 4$ encoder layers, a hidden size of $H = 512$, and $A = 8$ attention heads representing a total of 28M parameters[1]. Since this small language model is based on knowledge distillation, a fair description of this pre-training task is given next. Building a compact model revolves around knowledge distillation: the standard technique for model compression [7]. Since LLMs have a high computational cost, research on the development of a small model was guided by not only reducing the computational cost but also by using the same self-supervised learning paradigm in its development.

Indeed, building a compact model proved to be possible by applying the standard pre-training and fine-tuning process but a different training strategy, based on a compression technique known as knowledge distillation, was implemented. Basically, this distillation technique consists of a student-teacher training method where the teacher, a robust LM, transfers knowledge to the student, a small LM to be developed, through its predictions for unlabeled training examples. Fig. 1 shows the knowledge distillation process incorporated in the development and implementation of a compact BERT model [20]. The training resources demanded by the process are the following:

- Teacher: the teacher is a LLM which can be either a BERT-base or a BERT-large pre-trained language model.
- Student: the student is the compact model to be built. Whereas the total number of parameters is 110 million in BERT-base, the initial size for a tiny model is 4 million parameters.

Label data (D_L): a set of N training examples (x_1, y_1), . . ., (x_N , y_N), where x_i is an input and y_i is a label. Unlabeled training data (D_T): a set of M input examples $x_1^{'}$, ..., $x_M^{'}$ obtained from a distribution not necessarily identical to the distribution of

---

[1] _bert/bert_en_uncased_L-4_H-512_A-8

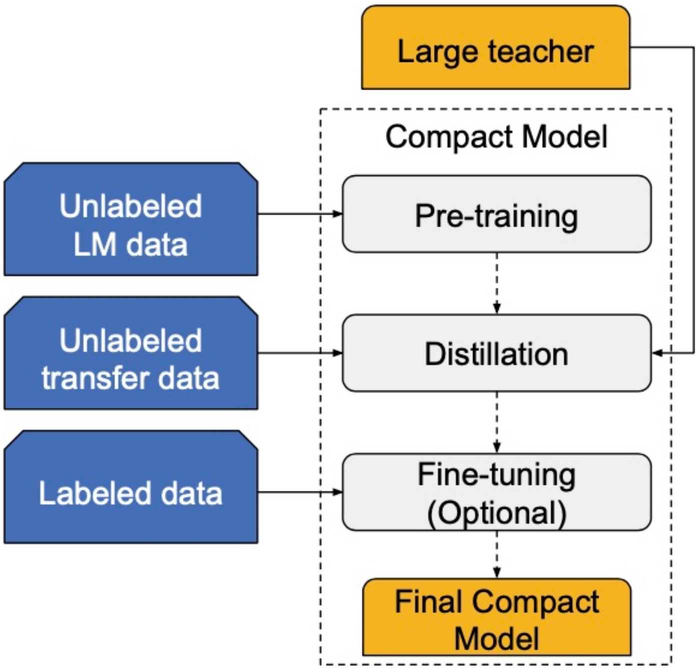


Fig. 1. Knowledge Distillation process. This figure corresponds to [20].

the labeled set. This dataset is used by the teacher for the transfer of knowledge to the student by making available its predictions for instances $x_m^{'}$.

Unlabeled language model data (D_{LM}): it is an unannotated text collection for unsupervised learning of text representation by using MLM as training method and the kernel of the compact model is defined for a sequence of three training operations: Pre-training on D_{LM}: pre-training of the compact model with MLM as training method. Distillation on D_T: transfer knowledge to the student. Once the student is prepared, the teacher transfer its knowledge to the student via its predictions to strengthen the compact model. The estimation of the cross-entropy loss between teacher and student predictions is then used to update the student model.

Fine-tuning on D_L: this operation is an optional step. The compact model is fine-tuned on end-task labeled data. In other words, the similarity between the distribution of the transfer and labeled datasets is perceived in this step. This compact model is compared with two contemporary works that also use distillation for transfer knowledge. Both works initialize the student with a BERT model truncated, that is, the bottom layers of a 12-layer BERT model are used for the initialization of the student. However, the distillation process is different.

Whereas Patient Knowledge Distillation performs task-specific distillation [17], DistillBert makes use of a more expensive LM teacher as distillation is performed on general-domain data [15].

4 Experimental Evaluation

The experimentation conducted in this work is the description of deep transfer learning for a particular text-processing task with recurrent neural networks architectures. So, we first explain the fine-tuning process of the pre-trained Compact model previously mentioned to perform a downstream task as sequence classification. As a fundamental part of the tuning process, the downstream network architectures implemented are also detailed. Then, the dataset characteristics are exposed and the results of each recurrent neural architectures are exhibited.

4.1 Fine-tuning

The process to make use of the generalizations produced by the pretrained language models is known as fine-tuning. These generalizations are helpful to build a sort of pipeline applications to cope with particular NLP tasks such as sequence classification or named entity tagging. In our classification experiments, we adopt two strategies to the downstream task: the use of the embeddings obtained from the pre-trained model as the input to a classic neural network, and a more refined optimization of such embeddings via deep recurrent neural networks.

So, how to obtain the embeddings from the compact BERT model? The BERT models return a map with three important keys: pooled_output, sequence_output and encoder_outputs. For the neural network architectures implemented in our tuning process, the first two keys are of interest to us:

- **pooled_output**: represents each input sequence as a whole so the embedding denotes the entire sequence. The shape is [batch_size, H].
- **sequence_output**: represents each input token in the sequence so in this case we have a contextual embedding for every token. The shape is [batch_size, seq_length, H].

In the case of pooled_output, Fig. 2 shows how there is an additional vector symbolized by the **[CLS]** token which is prepended to the input sequences. This additional vector [CLS] captures the entire sequence so it is provided to a classic neural network classifier that makes the category decision. By using a labeled dataset, the sequence classification task entails to learn a set of weights (W) in order to map the output vector (Y_{CLS}) to a set of categories:

$$y = \text{softmax}(WY_{\text{CLS}}). \tag{5}$$

On the other hand, the fine-tuning process with deep recurrent neural networks requires the use of sequence_output. Recurrent neural networks represent the temporal nature of language as each element of a sequence (each token in Fig. 2) is processed at a time [4].

The key point in this model is the computation of the hidden layer as the last input element denotes the entire sequence. To activate the current hidden layer is necessary the value obtained in the previous hidden layer corresponding to a preceding point in time. Equation (6) expresses the computation of the hidden layer **h** where **x** denotes

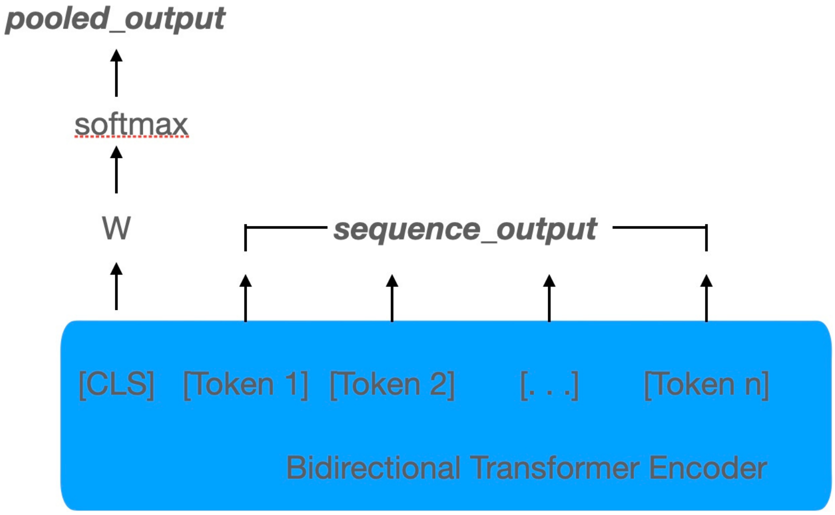


Fig. 2. BERT's output: pooled_output and sequence_output.

the sequence (i.e. input) and **g** an activation function. **W** denotes the weight matrix corresponding to the input x_t whereas **U** denotes the weight matrix corresponding to the hidden layer of the previous timestep h_{t-1}. In this way, this connectionist model is concerned with the context corresponding to each element of the sequence:

$$h_t = f(Uh_{t-1} + Wx_t). \tag{6}$$

4.2 Data

The dataset used in this experimentation is part of an ambitious research project denominated the Automated Student Assessment Prize (ASAP) [6] for automated grading of student-written responses sponsored by The William and Flora Hewlett Foundation. The purpose is to explore new forms of testing and grading methods and to reduce the cost of human graders by automating the student assessment. Three stages set up the ASAP project:

- Phase 1: analysis of essays: long-form response.
- Phase 2: analysis of short answers: short-form response.
- Phase 3: analysis of charts/graphs: symbolic mathematical/logical reasoning.

The focus of our attention is the collection of short-answers corresponding to the phase 2. Each instance in the collection denotes a short answer corresponding to a reading passage from a broad range of disciplines: From English Language Arts to Science. More specifically, the dataset is divided into 10 collections, where each one is described by a particular reading passage corresponding to a particular discipline and where the grade is defined in terms of levels of quality or categories: **0** (not proficient), **1** (partially proficient), or **2** (proficient).

The average length of each answer is approximately 50 words and most training sets contain around 1,800 responses that have been randomly selected from a sample of approximately 3,000. From the 10 training collections available in the dataset, we select four training sets where three levels of quality define the grade of each answer. In other words, the fine-tuning process implemented in our experimentation performs a downstream task as multi-class classification where a short answer is assigned into one of the multiple rubrics of the responses.

4.3 Results

As we previously said, we adopt ,in our classification experiments, two strategies to the downstream task: the use of the embeddings as the input to a classic neural network, and a more refined optimization of such embeddings via deep recurrent neural networks. Thus, the downstream network architectures implemented are:

- **Classic**: three dense layers are used to adjust the pre-trained embeddings obtained from pooled_output. The first and second layers contain 64 and 32 hidden units respectively, and since the number of hidden units of the compact BERT model is 512, and our experimentation performs a downstream three-class classification, the number of parameters to be adjusted is 35,011.
- **SimpleRNN**: as traditional neural networks are unable to represent the temporal nature of language [4], the embeddings are transferred step by step. A simple dense layer is used to adjust the pre-trained embeddings obtained from sequence_output. For example, since the number of hidden units of the compact BERT model is 512, a SimpleRNN layer with dimensionality of 50 (number of units), and a downstream three-class classification task, the number of parameters to be adjusted is 28,303.
- **LSTM**: as SimpleRNN cannot keep track of long-term dependencies, we try LSTM networks to include the consideration of distant constituents [8]. A simple dense layer is also used to adjust the pre-trained embeddings obtained from sequence_output. In this case, as the number of hidden units of the compact BERT model is 512, a LSTM layer with dimensionality of 50 (number of units), and a downstream three-class classification task, the number of parameters to be adjusted is 112,753.
- **Bidirectional**: as in many cases we need information from the context to the right of the current token, we try Bidirectional networks to include the consideration of the constituents from the start to the end of the input sequence and vice versa [16]. A simple dense layer is also used to adjust the pre-trained embeddings obtained from sequence_output. In this case, as the number of hidden units of the compact BERT model is 512, a Bidirectional network having LSTM layers with dimensionality of 50 (number of units), and a downstream three-class classification task, the number of parameters to be adjusted is 225,503.

As the size of the short-answers collections is small, the performance evaluation of the pre-trained models was conducted by the cross-validation method to use all the responses corresponding to a particular domain.

Table 1. ASAP-RNNs-50-units.

Architecture	**Dataset 3**	**Dataset 7**	**Dataset 8**	**Dataset 9**
Classic	0.71	0.66	0.64	0.72
SimpleRNN	0.79	0.79	0.78	0.83
LSTM	**0.84**	**0.84**	**0.82**	**0.86**
Bidirectional	0.84	0.84	0.82	0.86

Table 2. ASAP-RNNs-100-units.

Architecture	**Dataset 3**	**Dataset 7**	**Dataset 8**	**Dataset 9**
Classic	0.71	0.66	0.64	0.72
SimpleRNN	0.78	0.76	0.73	0.80
LSTM	**0.84**	**0.84**	**0.81**	**0.87**
Bidirectional	0.84	0.84	0.80	0.84

We train our downstream learning models with an Adam optimizer with a learning rate of 0.001, three-fold cross-validation and 25 epochs. We also apply dropout with $\rho = 0.2$ across layers of the downstream networks to prevent overfitting. Tables 1 and 2 show the results of the transfer of knowledge, obtained from the compact BERT model to deep recurrent neural networks. Table 1 shows the results for deep network architectures with a dimensionality of 50 units, whereas the results with a dimensionality of 100 units are exhibited in Table 2. The results are expressed in terms of the F1 score corresponding to each network architecture and each domain. For example, the second row of the Table 1 shows a F1 score of 0.79 obtained with a SimpleRNN architecture for dataset 3. A deep analysis of the results is carried out in the next section.

5 Discussion

A starting point for our discussion section is the definition of the baseline as a reference point for the obtained results. As it has been described in the data section, the data collection used in our experimentation is part of a competition for automated grading of student-written responses (ASAP) [6]. Unfortunately, the information available on the competition portal only mentions the winners of the competition but no methodology implemented or obtained results are provided.

However, taking into account that our purpose is to perceive the impact of deep transfer learning with recurrent neural networks, we define the classic neural network model as the baseline model. For the sake of clarity, Fig. 3 and Fig. 4 show a graphic perspective of the obtained results corresponding to the Tables 1 and 2 respectively.

Recurrent Neural Networks. Based on these figures, it is possible to observe whether there is a meaningful contrast in the use of different recurrent neural networks when transferring the generic knowledge produced by a pre-trained language model such as the compact BERT model.

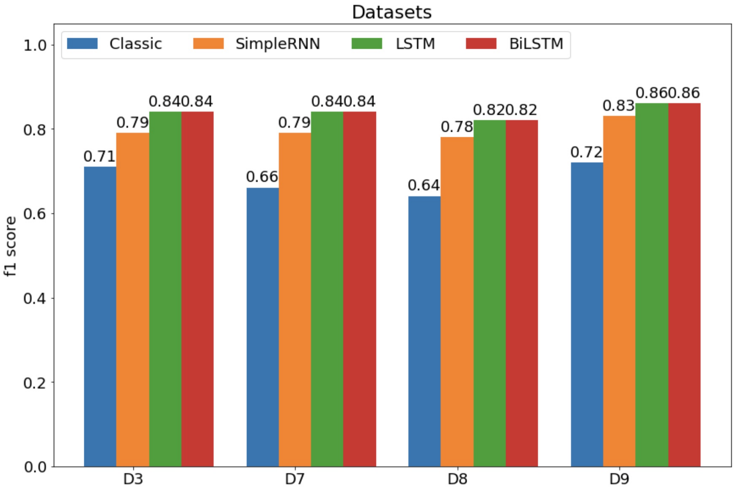


Fig. 3. ASAP-RNNs-50-units.

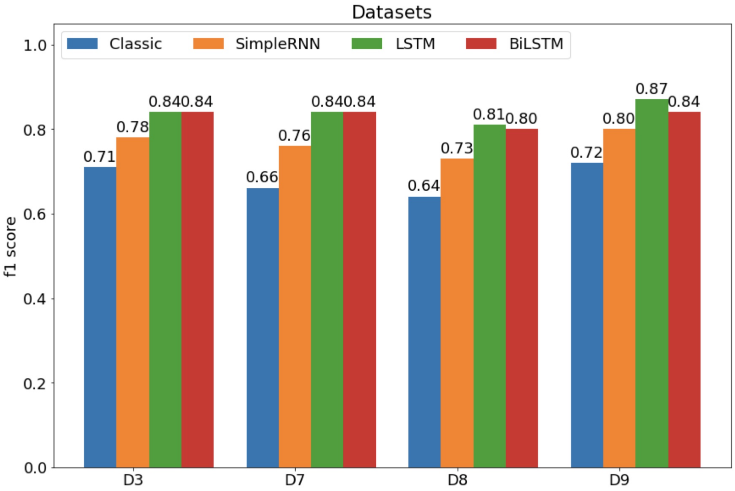


Fig. 4. ASAP-RNNs-100-units.

First of all, we see how the baseline performance denoted by a classic neural network model has been surpassed by the recurrent neural networks. A significant increase in the performance can be observed for all datasets. Now, the use of diverse recurrent neural networks exhibits important points to be noticed. First, we see how the transfer of the embeddings produced by the Compact model has been worth of implementing with recurrent neural networks. For all the observed datasets in Fig. 3 or Table 1, the F1 score obtained by the use of the SimpleRNN architecture is higher than the score obtained by the classic architecture. An average increase of 11 points in the F1 score is observed.

Fig. 3 or Table 1 also show how the highest performance has been obtained with the use of the LSTM architecture. For all the observed datasets, the transfer of knowledge with this recurrent architecture proved to be the best option. On the other hand, we see how the transfer of the embeddings has not been worth of implementing with a Bidirectional architecture such as BiLSTM (see Fig. 3 and Fig. 4). At its best score, this Bidirectional architecture achieves the same performance as the one obtained with the LSTM architecture. We attribute this result to the size of the texts: a short-answer text does not seem to demand information from the context to the right of the current token being analyzed.

Dimensionality. We have presented our results in two figures corresponding to the implementation of diverse recurrent neural networks with two dimensions: 50 and 100 units. We configured in this way taking into account that the average length of each answer is approximately 50 words. Regardless of the dataset observed, the F1 score obtained with an architecture of 100 units is equal or lower than the one obtained with an architecture of 50 units. Thus, the transfer of knowledge with a recurrent neural architecture of 50 units proved to be the best option.

A side effect of dimensionality is the increase in the number of parameters to be adjusted.For example, the number of parameters to be trained for a SimpleRNN architecture of 50 units is 28,303, whereas the number of parameters to be trained for a SimpleRNN architecture of 100 units is 61,603. And of course, given that the complexity of the LSTM and BiLSTM architectures is greater than the complexity of a SimpleRNN architecture, the number of parameters to be considered is much greater.

6 Conclusions

In this paper, we have analyzed the impact of deep transfer learning with RNNs. To consider similarities and differences between diverse RNNs, the transfer learning for each variant is described in terms of the fine-tuning process implemented on a downstream classification task. Our experimentation based on the classification of short-answer texts, provides empirical evidence of how the tuning of the embeddings obtained from a compact BERT model is worth of implementing with RNNs. Compared with a classic neural network, a SimpleRNN architecture improves the results, whereas the best results were obtained with the LSTM architecture.

References

1. Azunre, P.: Transfer learning for natural language processing. Manning Publisher (2021)
2. Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., Schwarzschild, A., Wilson, A. G., Geiping, J., Garrido, Q., Fernandez, P., Bar, A., Pirsiavash, H., LeCun, Y., Goldblum, M.: A cookbook of self-supervised learning (2023) doi: 10.48550/ARXIV.2304.12210
3. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186 (2019)

4. Elman, J. L.: Finding structure in time. Cognitive Science, vol. 14, no. 2, pp. 179–211 (1990) doi: 10.1016/0364-0213(90)90002-e
5. Goodfellow, I., Bengio, Y., Courville A.: Deep learning. The MIT Press (2016)
6. Hamner, B., Morgan, J., Lynnvandev., Shermis, M., Vander-Ark, T.: The Hewlett foundation: Automated essay scoring (2012) www.kaggle.com/competitions/asap-aes
7. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015) doi: 10.48550/arXiv.1503.02531
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation, vol. 9, no. 8, pp. 1735–1780 (1997) doi: 10.1162/neco.1997.9.8.1735
9. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A lite BERT for self-supervised learning of language representations. In: International Conference on Learning Representations, pp. 1–17 (2020)
10. Lin, Z., Feng, M., Nogueira-dos-Santos, C., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding (2017) doi: 10.48550/arXiv.1703.03130
11. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettle-Moyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach (2019) doi: 10.48550/arXiv.1907.11692
12. Logeswaran, L., Lee, H.: An efficient framework for learning sentence representations (2018) doi: 10.48550/ARXIV.1803.02893
13. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X.: Pre-trained models for natural language processing: A survey (2020) doi: 10.48550/arXiv.2003.08271
14. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research. vol. 21, no. 1, pp. 1–67 (2020)
15. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Smaller, faster, cheaper, lighter: Introducing DistilBERT, a distilled version of BERT (2019) doi: 10.48550/arXiv.1910.01108
16. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681 (1997) doi: 10.1109/78.650093
17. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for BERT model compression (2019) doi 10.48550/arXiv.1908.09355
18. Taylor, W. L.: Cloze procedure: A new tool for measuring readability. Journalism Quarterly, vol. 30, no. 4, pp. 415–433 (1953) doi: 10.1177/107769905303000401
19. TensorFlow: BERT (2020) tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-512_A-8/2
20. Turc, J., Chang, M. W., Lee, K., Toutanova, K.: Well-read students learn better: On the importance of pre-training compact models. In: Proceedings of the International Conference on Learning Representations, pp. 1–13 (2019) doi: 10.48550/arXiv.1908.08962
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1–11 (2017)
22. Wei, C., Xie, S. M., Ma, T.: Why do pretrained language models help in downstream tasks? An analysis of head and prompt tuning (2021) doi: 10.48550/ARXIV.2106.09226

ISSN 1870-4069

Development of an Autonomous Navigation and Manipulation Robot for Obstacle-Rich Environments

Andrea M. Salcedo-Vázquez, José A. León-Navarro,
Hortencia A. Ramírez-Vázquez, José G. Buenaventura-Carreón,
Fernando Antunez-Arnold

Tecnólogico de Monterrey,
Escuela de Ingeniería y Ciencias,
Monterrey, Mexico

{asalcedo2702,jantonioleon.irs}@gmail.com,
ha.ramirez@outlook.com,
{jgusbc,fer.antunez02}@hotmail.com

Abstract. This paper presents the development of an advanced autonomous navigation and manipulation robot specifically designed to operate efficiently in environments with numerous obstacles. The robot features a differential drive system, complemented by a camera and LiDAR sensors, which enable it to identify and locate target objects marked with ArUco codes. Leveraging advanced navigation algorithms and the Extended Kalman Filter (EKF) for sensor data fusion, the system achieves high-precision localization and calculates the target's position with notable accuracy. During operation, the robot dynamically plans its path to navigate toward the target while skillfully avoiding obstacles. Upon reaching the identified object, the robot uses a gripper to securely grasp the target and transport it to a designated unloading area. The integration of these functionalities underscores the robot's capability to perform complex tasks autonomously, significantly enhancing efficiency and safety in industrial environments. This research highlights the synergy of interdisciplinary technologies, combining robotics, sensor fusion, and control systems. Through rigorous iterative testing, the reliability and practicality of the system in real-world scenarios were validated. The proposed robotic solution offers a promising framework for automating tasks in complex environments, showcasing potential applications in various industries, including logistics and manufacturing, where precision and adaptability are critical.

Keywords: Differential mobile robot, navigation, manipulation, extended Kalman filter, optimization, autonomous.

1 Introduction

1.1 State of the Art in Mobile Robots

Mobile robots have significantly advanced, integrating sophisticated hardware and software to achieve high levels of autonomy and functionality.

pp. 49–62; rec. 2024-06-15; acc. 2024-08-17

Some of the most notable examples include Boston Dynamics Atlas, which utilizes high-performance hardware such as advanced actuators and sensors to perform complex tasks such as dynamic navigation and manipulation ([6]). Similarly, the TurtleBot series, commonly used in research and education, leverages ROS (Robot Operating System) and features hardware like the Intel RealSense camera and powerful processors to perform tasks such as mapping and human interaction ([10]).

Industrial autonomous mobile robots (AMRs) like KUKA's KMR iisy and KMP series incorporate AI and advanced path planning algorithms. These robots are equipped with high-end sensors, powerful on-board processors, and robust mechanical systems, enabling them to operate efficiently in dynamic and cluttered environments ([7]). For example, the KMR iisy includes an industrial-grade LiDAR, multiple cameras, and high-capacity batteries to support long operational hours.

In contrast, the Puzzlebot used in our research, developed by Manchester Robotics, operates with more constrained hardware resources. This robot is equipped with a Nvidia Jetson Nano featuring 2 GB of RAM, a Hackerboard for motor control, a Raspberry Pi Camera for vision, and a LiDAR sensor for obstacle detection. Despite these limitations, the Puzzlebot demonstrates the potential for effective autonomous navigation and manipulation through optimized algorithm implementation.

1.2 Contribution of This Paper

The advancements in mobile robotics are remarkable, but there is a critical challenge in adapting these sophisticated technologies to operate on less powerful hardware. This paper addresses this gap by optimizing various algorithms to work effectively on constrained hardware resources. Specifically, we adapted advanced navigation and manipulation algorithms to run on a Jetson Nano, a compact and less powerful computing module, as opposed to the high-performance GPUs typically used in state-of-the-art systems.

The primary contribution of this research is the demonstration of robust autonomous navigation and manipulation capabilities on a hardware platform with limited computational power. By fine-tuning the algorithms for real-time performance and efficient resource utilization, we showcase a cost-effective approach that broadens the accessibility and applicability of autonomous mobile robots in various fields, particularly where high-end hardware is not feasible.

This paper presents the development and implementation of an autonomous mobile robot equipped with a differential drive system, a camera, and LiDAR sensors. Using a suite of algorithms, including the Extended Kalman Filter (EKF) for sensor data fusion, the Bug0 algorithm for obstacle avoidance, and a proportional controller for path planning. The system's ability to navigate complex environments, identify and manipulate objects marked with ArUco codes, and perform tasks autonomously underscores its potential for industrial applications, particularly in settings requiring cost-effective and efficient robotic solutions.

By focusing on optimizing software to compensate for hardware limitations, this research contributes to the advancement of autonomous robotics, enabling the deployment of capable mobile robots in a wider range of scenarios, including educational, research, and industrial applications.

2 Puzzlebot

2.1 Hardware

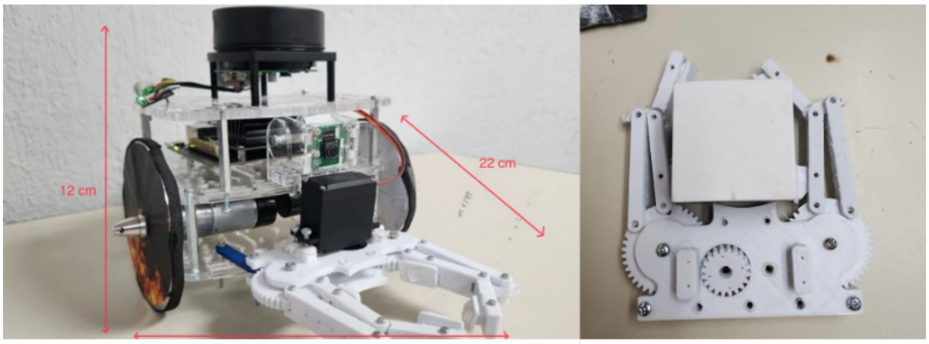
The Puzzlebot is an educational mobile robot designed for learning and experimentation in robotics. It features a compact and robust design with dimensions typically measuring around 22 cm in length, 17 cm in width, and 12 cm in height. This size makes it suitable for navigating and performing tasks in various environments, from classroom settings to more complex obstacle courses. (figure 1). The hardware of the Puzzlebot includes several key components.

- Jetson Nano, a powerful and versatile computing module from NVIDIA. The Jetson Nano provides the processing power necessary for running complex algorithms and machine learning tasks, making it possible for the Puzzlebot to perform real-time image processing and sensor data fusion.
- LiDAR sensor, the Puzzlebot can measure distances to surrounding objects with high precision. LiDAR works by emitting laser pulses and measuring the time it takes for the pulses to reflect back from objects, creating a detailed map of the environment. This sensor is crucial for obstacle detection and avoidance, enabling the robot to navigate safely through its environment.
- Hackerboard, a versatile microcontroller board that handles lower-level control tasks. The Hackerboard interfaces with various sensors and actuators, including the motors that drive the robot's wheels. It ensures smooth and precise control of the robot's movements, executing commands from the main processing unit (Jetson Nano) and managing the real-time operation of the robot's hardware components.

2.2 Software

On the software side, the Puzzlebot runs on an environment composed by different platforms:

- Linux-based operating system, leveraging the powerful capabilities of the Jetson Nano.
- ROS (Robot Operating System), which provides a flexible framework for writing robot software. ROS offers tools and libraries for handling sensor input, motion control, and inter-component communication, streamlining the development process.
- For Simulation, we implemented different softwares as Gazebo and RViz:
  - Gazebo, a robust robotics simulator that allows us to create realistic 3D environments where the Puzzlebot can be tested. Gazebo helps in testing and refining algorithms before deploying them on the physical robot, reducing development time and ensuring safer initial trials.
  - RViz, a visualization tool that works with ROS to visualize sensor data and the robot's state in real-time. RViz enables us to monitor the robot's perception of its environment and its planned path, providing valuable insights into its behavior and performance.



Puzzlebot used in real-life

Gripper

Fig. 1. Puzzlebot an gripper.

3 Mathematical Modeling

3.1 Reference Frames Definition

To have all the elements of the system in the same coordinate frame, we define our fixed frame, which in this case we call ”odom”. We perform a transform from the robot’s base to ”odom” to obtain the coordinates of the Puzzlebot with respect to the ”odom” reference frame. Similarly, we need to perform a transform from the LiDAR to the robot’s base. This way, the point cloud shown by the LiDAR moves as the robot moves, allowing us to map our environment and determine the distances to obstacles. Finally, we perform a transform from Aruco to the center of the camera and from the camera to the base. This allows us to know the position of the Aruco with respect to the robot and its location within the ”odom” frame.

3.2 Kinematic Model of a Differential Robot

A differential drive robot has two independently driven wheels mounted on either side of the robot, which allows it to move and steer by varying the relative speeds of the wheels. The kinematic model describes the relationship between the wheel speeds and the robot’s overall movement.

Equations. The linear and angular velocities of the robot are related to the wheel velocities as follows:

$$v = \frac{v_r + v_l}{2} = r\frac{w_r + w_l}{2}, \tag{1}$$

$$w = \frac{v_r - v_l}{2} = r\frac{w_r - w_l}{l}. \tag{2}$$

The robot’s pose (position and orientation can be represented by (x, y, θ), where:

- (x, y) is the position of the robot in the plane odom.
- θ is the orientation of the robot with respect to the x-axis of odom.

Kinematic Equations. The Kinematic equations that describes the Puzzlebot motion are:

$$\dot{x} = v\cos(\theta), \tag{3}$$

$$\dot{y} = v\sin(\theta), \tag{4}$$

$$\dot{\theta} = w. \tag{5}$$

These equations can be integrated over time to simulate the robot's trajectory given the angular wheel velocities w_r and w_l using the robot encoders. Getting as result:

$$x_k = x_{k-1} + v\cos(\theta) \times dt, \tag{6}$$

$$y_k = y_{k-1} + v\sin(\theta) \times dt, \tag{7}$$

$$\theta_k = \theta_{k-1} + w \times dt. \tag{8}$$

4 Navigation

4.1 Robot Navigation Algorithms

For the robot's navigation, we used two main algorithms: proportional controller to reach a set point, and bug 0 to avoid obstacles [4].

- **Proportional Controller.** We utilized a proportional controller to facilitate movement from point A to point B. This controller regulates the Puzzlebot's linear and angular velocities by calculating the distance an angle error between the starting an destination points. The P controller ensures that the robot can efficiently reach its target by adjusting it speeds based on the proximity and orientation to the goal.
- **Bug0.** The Bug0 algorithm is activated when the Puzzlebot detects an obstacle, such as a wall, using its LiDAR sensor at a minimum distance:

Algorithm 1 Bug 0 algorithm.

```
if Wall then
    Turn 90 degrees
    while not wall do Follow Wall
        if angle aligns with the target point then
            Go to point
            break
        end if
    end while
end if
```

4.2 State Machine

A state machine was designed to control the Puzzlebot actions to guide its behavior through different tasks:

- **Start/Stop:** The robot starts in the initial state, waiting for the ID of the cube and the base where it should be dropped off.
- **Search and pick up the cube:** Upon receiving the cube ID and base location, the robot transitions to searching for the cube. Once the cube is found, the robot approaches and grabs it, switching to the localization state.
- **Localization and Navigation:** In the localization state, the robot activates the P controller to navigate towards the target point. If a wall is encountered, the Bug0 algorithm is triggered, guiding the robot around obstacles until it is aligned to move directly towards the point.
- **Go to base:** Upon reaching the desired point, the robot changes to the state of searching for the base. Once the Aruco marker identifying the base is detected, the robot approaches it, places the cube, and returns to the initial state.

5 Localization

5.1 Odometry

Odometry is a method used to estimate the position and orientation of a mobile robot by tracking the motion of its wheels or other actuators. This technique involves measuring the distance traveled by each wheel and using these measurements to calculate the robot's change in position over time. By continuously monitoring the wheel encoder data, the robot can estimate its current position and orientation as it moves through the environment. However, odometry has its challenges; it is prone to cumulative errors due to wheel slippage, uneven surfaces, and other factors that can introduce inaccuracies over time.

5.2 Equations

To calculate the odometry pose we use the equations 6, 7 and 8. These equations allow us to update the robot's position and orientation over time. Additionally, we utilized ROS2 odom message to publish the odometry data to other nodes. This enables real-time visualization in RViz, where the angle is sent in a quaternion format to ensure accurate orientation representation.

5.3 Map Based Localization

For localization, we relied on map-based techniques to determine the robot's position and orientation using odom frame as out map. We match the sensor data as the LiDAR distances or the ARUCO position to accurately localize itself and navigate effectively in its environment.

To mitigate these errors, odometry is often combined with other sensors and algorithms, such as the Extended Kalman Filter (EKF) and LiDAR, to improve the overall accuracy of the robot's position estimate. This sensor fusion approach leverages the strengths of each sensor, providing a more robust and reliable navigation solution. The integration of odometry with the EKF and other sensory inputs allows the Puzzlebot to navigate autonomously with high precision, even in complex and dynamic environments.

6 Extended Kalman Filter (EKF)

The implementation of the EKF in our system enhances the robot's ability to navigate autonomously and avoid obstacles with high precision. By continuously refining the state estimate, the EKF ensures that the robot's control algorithms receive accurate and up-to-date information, enabling smooth and efficient navigation. This makes the EKF an essential component in achieving robust performance in complex and dynamic environments [3].

6.1 System Linearization and Covariance Matrix

To enhance the accuracy of our odometry, we performed a linearization of the system. This involves approximating the nonlinear system around a specific operating point to simplify the analysis and control design.

Linearization Process. The kalman filter is a very powerful tool optimal for linear systems with gaussian noise. Hoewever in real world there are many non-linear systems, such as the differential mobile robots, which is whu we must linearize the system to apply the kalman filter. The linearization of our system was made by calculating the Jacobian matrices of our systems. We calculate two matrices, one for the position of the robot in the axis x, y and z (equation 10) and another one for the linear and angular position (equation 11) making partial derivates of the system with respect to each of the system's states. In the equation 12 we can see the linearized model:

$$\dot{z} = \begin{cases} \dot{x} = v\cos(\theta), \\ \dot{y} = v\sin(\theta), \\ \theta = w, \end{cases} \tag{9}$$

$$A = \begin{bmatrix} \frac{dV\,\mathrm{Cos}(\theta)}{dx} & \frac{dV\,\mathrm{Cos}(\theta)}{dy} & \frac{dV\,\mathrm{Cos}(\theta)}{d\theta} \\ \frac{dV\,\mathrm{Sin}(\theta)}{dx} & \frac{dV\,\mathrm{Sin}(\theta)}{dy} & \frac{dV\,\mathrm{Sin}(\theta)}{d\theta} \\ \frac{dw}{dx} & \frac{dw}{dy} & \frac{dw}{d\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -V\,\mathrm{Sin}(\theta) \\ 0 & 0 & V\,\mathrm{Cos}(\theta) \\ 0 & 0 & 0 \end{bmatrix}, \tag{10}$$

$$B = \begin{bmatrix} \dfrac{dV\,\mathrm{Cos}(\theta)}{dv} & \dfrac{dV\,\mathrm{Cos}(\theta)}{dw} \\ \dfrac{dV\,\mathrm{Sin}(\theta)}{dv} & \dfrac{dV\,\mathrm{Sin}(\theta)}{dw} \\ \dfrac{dw}{dv} & \dfrac{dw}{dw} \end{bmatrix} = \begin{bmatrix} \mathrm{Cos}(\theta) & 0 \\ \mathrm{Sin}(\theta) & 0 \\ 0 & 1 \end{bmatrix}, \tag{11}$$

$$\dot{z} = \begin{bmatrix} 0 & 0 & -V\,\mathrm{Sin}(\theta) \\ 0 & 0 & V\,\mathrm{Cos}(\theta) \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \mathrm{Cos}(\theta) & 0 \\ \mathrm{Sin}(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}. \tag{12}$$

Covariance Matrix Calculation. The covariance matrix Σ_k is crucial for estimating the uncertainty in the robot's position and orientation. It evolves over time based on the system dynamics and measurement updates. It calculation is defined by:

$$\Sigma_k = H_k \Sigma_{k-1} H_k^T + Q_k, \tag{13}$$

where:

- Σ_k is a 3x3 covariance matrix:

$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} \end{bmatrix}. \tag{14}$$

- H_k is a 3x3 Linear model Jacobian of the robot:

$$\begin{bmatrix} 1 & 0 & -\delta t \cdot v_k \cdot \sin(\mu_{\theta,k-1}) \\ 0 & 1 & \delta t \cdot v_k \cdot \cos(\mu_{\theta,k-1}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{15}$$

- Q_k matrix is the nondeterministic error matrix, given by:

$$Q_k = \nabla_{w_k} \cdot \Sigma_{\delta,k} \cdot \nabla_{w_k}^T, \tag{16}$$

where:

$$\Sigma_{\delta,k} = \begin{bmatrix} k_r|w_{r,k}| & 0 \\ 0 & k_l|w_{l,k}| \end{bmatrix}, \tag{17}$$

$$\nabla_{w,k} = \frac{1}{2} r \delta t \begin{bmatrix} \cos(s_{\theta,k-1}) & \cos(s_{\theta,k-1}) \\ \sin(s_{\theta,k-1}) & \sin(s_{\theta,k-1}) \\ \frac{2}{l} & -\frac{2}{l} \end{bmatrix}. \tag{18}$$

6.2 Kalman Filter Algorithm and Equations

For the implementation of the kalman filter, we followed these steps:

- **Position Estimation:** Initially, we estimated the position using odometry calculated from the encoder data. This provided us with an initial estimate of the robot's location and we called it $\hat{\mu}_k$.
- **Linearized model and uncertainty:** Calculate the linearized model (equation 15) to calculate the uncertainty propagation which is calculated by:

  $$\hat{\Sigma}_k = H_k \cdot \Sigma_{k-1} \cdot H_k^T + Q_k, \tag{19}$$

  where Q_k is the motion model covariance matrix
- **Observation model:** If the Puzzlebot could detect the Aruco marker with the camera, we calculated the distance and angle from the Puzzlebot to the Aruco marker using the camera data which will be our assumed measurements $\hat{z}$. And then we construct the observation model by calculating the distance and angle from the puzzlebot to the aruco marker using the aruco and robot positions:

  $$\hat{z}_k = \begin{bmatrix} \hat{z}_{p,k} \\ \hat{z}_{\theta,k} \end{bmatrix} = \begin{bmatrix} \sqrt{\delta x^2 + \delta y^2} \\ \operatorname{atan} 2(\delta y, \delta x) - \hat{s}_{\theta,1} \end{bmatrix}. \tag{20}$$

- **Observation Model Linearization:** The observation model was linearized to facilitate the estimation process. This step involved computing the Jacobian matrix of the observation model:

  $$G_k = \begin{bmatrix} -\frac{\delta x}{\sqrt{p}} & -\frac{\delta y}{\sqrt{p}} & 0 \\ \frac{\delta y}{p} & -\frac{\delta x}{p} & -1 \end{bmatrix}. \tag{21}$$

- **Uncertainty Propagation and Kalman Gain Calculation:** We measured the propagation of uncertainty and calculated the Kalman gain. The Kalman gain determines how much weight is given to the prediction and observation updates in the estimation process.

- **Measurement Uncertainty:**

$$Z_k = G_k \cdot \hat{\Sigma}_k \cdot G_k^T + R_k, \tag{22}$$

where, R_k is the observation model covariance matrix.

- **Kalman Gain:**

$$K_k = \hat{\Sigma}_k \cdot G_k^T \cdot Z_k^- 1. \tag{23}$$

Position and Covariance Estimation Finally, we used the Kalman filter to estimate the position and covariance of the Puzzlebot. This iterative process helped us reduce uncertainty and improve the accuracy of the estimated position.

- **Robot position:**

$$\mu_k = \hat{\mu}_k + K_k(z_k - \hat{z}_k). \tag{24}$$

- **Covariance:**

$$\Sigma_k = (I - K_k \cdot G_k) \cdot \hat{\Sigma}_k. \tag{25}$$

7 Vision Algorithm

The use of ArUco markers involves attaching them to target objects that the Puzzlebot needs to identify, approach, and manipulate. By leveraging the ArUco detection library, the robot can accurately determine the position of these markers in real-time, enabling it to perform complex tasks such as picking up and placing objects at designated locations. This technology enhances the robot's ability to interact with its environment in a controlled and predictable manner, ensuring successful task execution.

7.1 Algorithm

For Aruco detection, we utilized a ROS2 library called ros_aruco_opencv. This library provides us with the Aruco ID and the position (x, y, z, theta) relative to the camera frame. To integrate this information into our navigation system, we initially perform a coordinate transformation from the camera frame to the base_link and then to the odometry frame. This allows us to determine the Aruco's position in the map coordinate system. Subsequently, we activate the proportional controller to guide the Puzzlebot towards the Aruco marker, positioning it for cube manipulation tasks.

We used different ArUco markers for different purposes, such as identify different objects and places or to apply kalman filter so the robot's position is more precise. The ArUcos we used are generated in a 4x4 dictionary and a size of 45 mm. We used ArUco ids 1, 2 and 3 to identify the stations where the object must be dropped, the id 6 to identify the cube the robot must grab and the id 7 to apply kalman filter.

8 Manipulator

Drawing inspiration from various online resources and videos, we a suitable gripper model on Thingiverse designed for robots participating in the FIRST Tech Challenge competition. The chosen model served as a foundational template, which we adapted to align with the specific requirements and constraints of the tests. Adjustments were made to accommodate the dimensions and shapes of the ArUco markers, ensuring a snug and secure grip during transportation and positioning tasks.

9 Results

This section presents the results of our autonomous navigation and manipulation robot's performance, comparing its behavior in simulated and real-world environments. Our primary focus is on the robot's stopping distance and the implications of the differences observed due to varying processing capabilities.

9.1 Simulation Performance

In the simulated environment, we employed a high-performance Nvidia RTX 4060 Ti GPU with 4352 CUDA cores and 16 GB of RAM. This powerful setup allowed for rapid and precise computations, enabling the robot to stop accurately at a distance of 30 centimeters from detected obstacles. The simulation environment, managed using ROS 2 and Gazebo, provided a controlled setting where sensor data processing and algorithm execution could occur without significant latency.

9.2 Real-world Performance

Conversely, the real-world tests were conducted using a Jetson Nano with 2 GB of RAM and a GPU featuring only 128 cores. This considerable disparity in processing power impacted the robot's performance, most notably in its stopping distance. The real-world robot consistently crash into the wall and we need to adjust the values so it stopped at 30 centimeters from obstacles, highlighting a discrepancy from the simulation results.

9.3 Analysis of Discrepancies

The observed difference in stopping distances—30 centimeters in simulation versus 30 centimeters in the real world—can be attributed to several factors:

- **Processing Capability:** The Jetson Nano's limited processing power affected real-time sensor data processing and decision-making speed. The RTX 4060 Ti's superior computational capacity allowed for faster and more precise calculations, resulting in a more accurate stopping response.
- **Sensor Data Throughput:** In the simulation, data throughput and processing are optimized by the high-performance GPU, reducing latency in sensor fusion and decision-making processes. In contrast, the Jetson Nano's limited throughput capacity led to slower data processing, contributing to the reduced stopping distance.

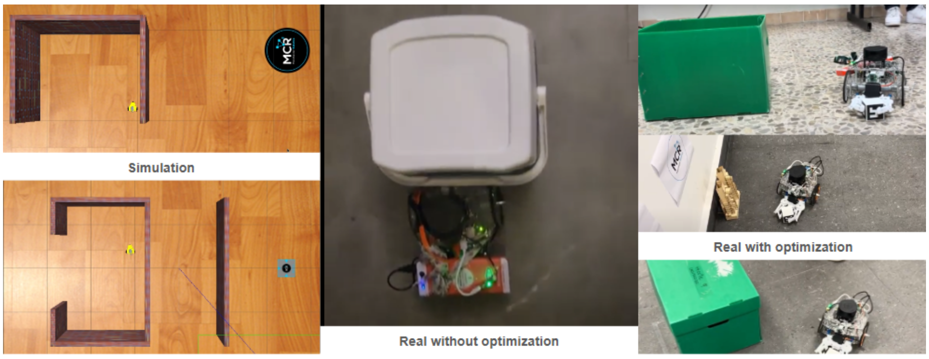


Fig. 2. Result of the tests.

- **Algorithm Efficiency:** The real-world implementation faced practical challenges such as sensor noise and physical limitations, which were less pronounced in the controlled simulation environment. These real-world factors necessitated adjustments to the robot's algorithms, further affecting its stopping distance.
- **Environmental Variability:** The real-world environment introduces variability in sensor readings due to factors such as lighting conditions, surface textures, and physical obstacles, which are typically idealized in simulations.

9.4 Performance Implications

The discrepancy between the simulation and real-world performance underscores the importance of considering hardware limitations and environmental factors when deploying robotic systems. While simulations provide valuable insights and a controlled platform for initial algorithm testing, real-world trials are crucial for fine-tuning and validating the robot's performance. Once we identified the discrepancies between the performance in simulation and in the real-world scenarios, we decided to optimize the sensor usage on the robot. Instead of activating all sensors simultaneously, we configured the robot to utilize only the necessary sensors for each specific state of its operation. This approach ensured that the robot could efficiently perform its tasks without overwhelming its limited processing capacity.

Additionally, we reduced the data processing load by minimizing the number of messages being sent and received. By streamlining the communication and focusing on essential data, we achieved a more efficient and responsive system, enhancing the robot's overall performance in real-world applications. Furthermore, the implementation of the Kalman filter and ArUco marker detection significantly improved the precision of the real robot. This enhancement was evident when comparing the simulation results with the real-world outcomes, both before and after optimization 2.

10 Resources

- Demonstrative video (Puzzlebot final challenge): youtu.be/YZyWaMJPywo?si=ayveFs6nnmhTK7tX

- GitHub Repository (Puzzlebot algorithms): github.com/soyhorteconh/Puzzlebot_Lidar_ROS1_ROS2
- Flow Diagrams Folder: drive.google.com/drive/folders/1k8Eu3JkrMkmCKipy-2vJNTDl004k5s8H?usp=sharing

11 Conclusions

The research presented has successfully developed an autonomous navigation and manipulation robot tailored for obstacle-rich environments. By integrating advanced sensors, such as LiDAR and cameras, with robust navigation algorithms, the robot demonstrated efficient and precise operations in complex scenarios.

The use of the Extended Kalman Filter significantly improved the accuracy of the robot's localization, enhancing its ability to navigate and manipulate objects. The implementation of ArUco markers for object identification and target localization proved to be effective, allowing the robot to perform tasks with a high degree of accuracy. Additionally, the differential drive system, combined with the Bug0 and proportional control algorithms, enabled reliable obstacle avoidance and target acquisition. This study underscores the potential of autonomous robots in industrial applications, where they can enhance operational efficiency and safety.

The integration of interdisciplinary technologies and the iterative refinement of both hardware and software components were crucial to the project's success. The findings from this research contribute to the broader field of robotics, offering insights into the development of more advanced and capable autonomous systems for future applications.

References

1. Borenstein, J., Everett, H. R., Feng, L., Wehe, D.: Mobile robot positioning: Sensors and techniques. Journal of Robotic Systems, vol. 14, no. 4, pp. 231–249 (1997)
2. Dudek, G., Jenkin, M.: Computational principles of mobile robotics. Cambridge University Press (2010)
3. Li, Q., Li, R., Ji, K., Dai, W.: Kalman filter and its application. In: Proceedings of the 8th International Conference on Intelligent Networks and Intelligent Systems, pp. 74–77 (2015) doi: 10.1109/icinis.2015.35
4. Lumelsky, V. J., Stepanov, A. A.: Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. Algorithmica, vol. 2, no. 1–4, pp. 403–430 (1987) doi: 10.1007/bf01840369
5. Luo, R., Kay, M.: Multisensor integration and fusion in intelligent systems. IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, no. 5, pp. 901–931 (1989) doi: 10.1109/21.44007
6. Parker, L. E.: Current state of the art in distributed autonomous mobile robotics. Distributed Autonomous Robotic Systems, pp. 3–12 (2000) doi: 10.1007/978-4-431-67919-6_1
7. Sharma, N., Pandey, J. K., Mondal, S.: A review of mobile robots: Applications and future prospect. International Journal of Precision Engineering and Manufacturing, vol. 24, no. 9, pp. 1695–1706 (2023) doi: 10.1007/s12541-023-00876-7
8. Siciliano, B., Khatib, O.: Springer handbook of robotics. Springer, 2nd Ed. (2016)

9. Siegwart, R., Nourbakhsh, I. R., Scaramuzza, D.: Introduction to autonomous mobile robots. MIT Press (2011)
10. Tagliavini, L., Colucci, G., Botta, A., Cavallone, P., Baglieri, L., Quaglia, G.: Wheeled mobile robots: State of the art overview and kinematic comparison among three omnidirectional locomotion strategies. Journal of Intelligent and Robotic Systems, vol. 106, no. 3 (2022) doi: 10.1007/s10846-022-01745-7
11. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. MIT Press (2005)

ISSN 1870-4069

Enhancing Music Genre Classification Using Tonnetz and Active Learning

Omar Velázquez-López[1], José Luis Oropeza-Rodríguez[1],
Gibran Fuentes-Pineda[2]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] Universidad Nacional Autónoma de México,
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas,
Mexico

{ovelazquezl2018, joropeza}@cic.ipn.mx,
{gibranfp}@unam.mx

Abstract. The task of music genre recognition (MGR) has significant applications in the music industry, including copyright control and music recommendations. Traditional feature engineering approaches in classical machine learning (ML) have utilized spectral features like Mel Frequency Cepstral Coefficients (MFCC), alongside time and frequency domain features such as zero-crossing rate, spectral centroid, and spectral rolloff. This study investigates the impact of harmonic, tonal, and rhythmic features, specifically tonnetz, chroma, and tempo, on the performance of ML models. We implemented Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and XGBoost, both in their traditional forms and with the integration of active learning, to classify genres in the GTZAN dataset. Our results demonstrate that active learning significantly improves model accuracy, with the highest accuracy achieved by the active SVM model at 80.3% using a combination of tonnetz and chroma features. This study underscores the importance of tonal and rhythmic features, particularly tonnetz, in optimizing MGR models. Future work includes expanding the dataset using the developer API from a music streaming platform and exploring alternative feature representations such as auditory filter banks.

Keywords: music genre classification, active learning, tonnetz, chroma, tempo, support vector machine, k-nearest neighbors, xgboost, GTZAN dataset, music streaming platform API.

1 Introduction

The music genre recognition (MGR) has been a fundamental task in the field of music recognition and information retrieval since the early 21st century, with works by authors such as Pachet and Cazaly [1] and Tzanetakis and Cook [2]. Currently, this task has significant applications in the music industry, such as copyright control, genre classification, and music recommendations, driven by the rise of music streaming platforms [3]. In classical machine learning (ML), feature engineering plays a crucial

pp. 63–73; rec. 2024-06-21; acc. 2024-08-21

role in measuring the results and performance of models, differing from deep learning algorithms where features are typically extracted automatically during the training process.

To address the problem of MGR, spectral features such as Mel Frequency Cepstral Coefficients (MFCC), along with time and frequency domain features like zero-crossing rate, spectral centroid, and spectral rolloff, have traditionally been used.

This article proposes that harmonic, tonal, and rhythmic features also significantly influence the representation of the musical signal and, consequently, the model's performance. We use tonnetz as a harmonic feature, chromagram as a tonal feature, and tempo (beats per second) as a rhythmic feature. The tonnetz is a visual representation of the harmonic relationships between chords and notes in music, while the chroma features represents the energy of the 12 musical notes over time. It is important to mention that some of the most significant works in chord recognition have used chroma [4] and tonnetz [5] features.

This study focuses on exploring and demonstrating the relevance of tonnetz in MGR using classical machine learning models. We hypothesize that tonnetz is the prominent musical feature in genre classification, as chord progressions tend to vary more between genres than chromatic tones or song tempos. For example, a jazz song and a rock song may share similar chromatic tones and tempos but will have distinctive chord progressions. To this end, we implement Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and XGBoost, both in their traditional form and by adapting the active learning method proposed by Deng and Ko [6], to classify genres in the GTZAN dataset.

2 Related Work

In the state-of-the-art of musical genre classification (MGR), traditional audio features such as Mel Frequency Cepstral Coefficients (MFCC) and time and frequency domain scalar features, such as zero-crossing rate, spectral centroid, and spectral rolloff, have been widely used in various studies. These audio features have proven effective in representing the acoustic information of music signals.

For instance, Qi et al. [7] include features such as spectral centroid, spectral rolloff, spectral flux, zero-crossing rate, and low energy in their analysis. Ghildiyal et al. [8] use zero-crossing rate, Root Mean Square Energy (RMSE), MFCC, spectral centroid, and spectral rolloff. Elbir et al. [3] employ zero-crossing rate, spectral centroid, spectral contrast, spectral bandwidth, and spectral rolloff. However, Qi et al. [7] and Ghildiyal et al. [8] also mention using musical features. Qi et al. [7] mention features related to texture, timbre, and instrumentation without specifying exactly which features, while Ghildiyal et al. [8] specify including chroma features. Table 1 provides a summary of the features used in each study according to their category.

Both Qi et al. [7] and Ghildiyal et al. [8] calculate two statistical measures for each feature, while Elbir et al. [3] add the standard deviation in addition to these two measures, doubling or tripling the feature size. This is the reason why the feature vectors in each work reach those approximately sizes.

Table 1. Features used in musical genre classification.

Categ.	**Feature**	**Elbir et al. [3]**	**Qi et al. [7]**	**Ghildiyal et al. [8]**
Traditional	MFCC	yes	yes	yes
	Zero Crossing Rate	yes	yes	yes
	Spectral Centroid	yes	yes	yes
	Spectral Rolloff	yes	yes	yes
	Spectral Contrast	yes	-	-
	Spectral Bandwidth	yes	-	-
	Flux	-	yes	-
	Low Energy	-	yes	-
	RMSE	-	-	yes
Musical	Chroma	-	not specified	yes
	Tonnetz	-	not specified	-
	Tempo	-	not specified	-
	Vector size (approx.)	93	60	72

Table 2. Some results for GTZAN in state of art.

Reference	**Model**	**Accuracy %**
Qi et al. [7]	KNN	90
Ghildiyal et al. [8]	Decision Tree	74.3
Elbir et al. [3]	SVM	72.7

Regarding machine learning models, techniques ranging from classical ML, such as k-nearest neighbors (KNN) and support vector machines (SVM), to deep learning (DL) models like convolutional neural networks (CNN) have been applied in MGR.

In the work by Elbir et al. [3], ML models from KNN to SVM and a DL model (CNN) trained with other spectrogram features were trained and evaluated. On the other hand, Ghildiyal et al. [8] compared several ML models, including Decision Tree, Random Forest (RF), and KNN. Both studies trained and evaluated their models' using samples from the GTZAN dataset. Table 2 shows the best-performing model from each work cited in this section. We have only considered results where ML algorithms were evaluated with the GTZAN dataset.

In Table 2 it is also important to note that the high performance of Qi et al.'s [7] KNN model is likely due to the fact that their work not only trained the model with samples from GTZAN, as Ghildiyal and Elbir did, but also added nearly 18,000 samples obtained from the Spotify developer API.

Furthermore, it is important to mention that Ghildiyal et al. [8] achieved even higher performance using their DL model. However, for the purposes of this work, which focuses on ML algorithms, it is not comparable. A similar situation occurs with Deng's work [6], which, although their contribution of active learning during model training is valuable to this work, evaluates their model with a different dataset.

3 Methodology

3.1 Dataset

The GTZAN dataset is widely used for music genre classification (MGC). It consists of 1000 audio tracks, each 30 seconds long, divided into ten genres: Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae, and Rock [3]. Its small size aligns well with the focus of this paper, as applying ML algorithms does not require the large amount of data needed by DL algorithms.

3.2 Preprocessing and Features

Data preprocessing includes several stages:

- MFCC Extraction: 20 MFCC coefficients were extracted from each audio sample to capture frequency characteristics with half-second windows.
- Time and Frequency Domain Features: zero-crossing rate, spectral centroid, and spectral rolloff were calculated to obtain additional information from the audio.
- Musical Features: Features such as tonnetz, chroma, and tempo were added to evaluate their impact on genre classification.

Considering that the features shared by all works in Table 1 are MFCC, zero-crossing rate, spectral centroid, and spectral rolloff, these were extracted and feature vectors were created as shown in Table 3. In this work, we decided to call them base features, as they will be used as the minimum length of the feature vector. On the other hand, Table 4 shows the respective size for each musical feature.

The size of the feature vector in each experiment will depend on which musical feature is added. Thus, it can range from a length of 46 to a maximum of 62.

To illustrate the musical features, we generated tonnetz spectrograms, chromagram, and tempo (Beats) plots for representative examples of the jazz and rock genres. Figure 1 shows these plots for the track 'jazz.00004', while Figure 2 presents the same features for the track 'rock.00009'. The tonnetz spectrograms provide a visual representation of the harmonic relationships between chords and notes in music, the chromagram represents the energy of the 12 musical notes over time, and the tempo plots show the onset strength and the distribution of beats over time.

Table 3. Base features.

Feature	Statistical measure	# of features
MFCC (20 coeff)	Mean and variance	40
Zero Crossing Rate		2
Spectral Centroid		2
Spectral Rolloff		2
Vector size		46

Table 4. Musical features.

Feature	Statistical measure	# of features
Tonnetz (6 relationships)	Mean and variance	12
Chroma (12 tones)		24
Tempo (unique value)		2
Vector size		16

The visualizations in Figures 1 and 2 demonstrate how chord progressions and tonal and rhythmic features vary between musical genres. In the tonnetz spectrogram, we can observe significant differences in the harmonic relationships between the jazz and rock tracks, highlighting the tonnetz's ability to capture the distinctive harmonic structure of each genre.

Although the chromagram appears to reveal significant differences in this instance, this is likely because the two tracks are in different musical scales. If both tracks were in the same scale, despite being different genres, they might risk appearing similar, as would be the case with the execution speed, i.e., the tempo.

These differences in tonal and rhythmic features provide a basis for assuming they can aid in music genre classification. However, this exercise is purely intuitive, so it will be necessary to measure the mentioned differences using ML.

3.3 Algorithms and Experiments

In this section, we detail the training procedure for both traditional and active learning approaches using ML algorithms. For all these models, we used numpy files containing manually extracted features from the audio signals. The data was pre-scaled.

The dataset was divided into training and testing sets, with a 70% training and 30% testing split. For each model, hyperparameter optimization was performed through grid search, selecting the best parameters using scikit-learn, except for XGBoost, which was built using the xgboost library.

For each model presented in this paper, hyperparameter optimization was performed through grid search, selecting the best parameters using scikit-learn, except for XGBoost, which was built using the xgboost library.

a) SVM: The optimization of this model consisted of finding the best hyperparameters C and gamma. We fixed an RBF kernel due to its ability to

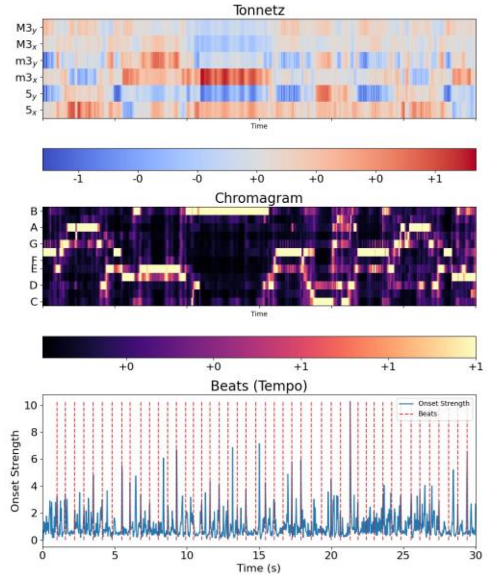


Fig. 1. Tonnetz spectrogram, chromagram and tempo of a jazz track.

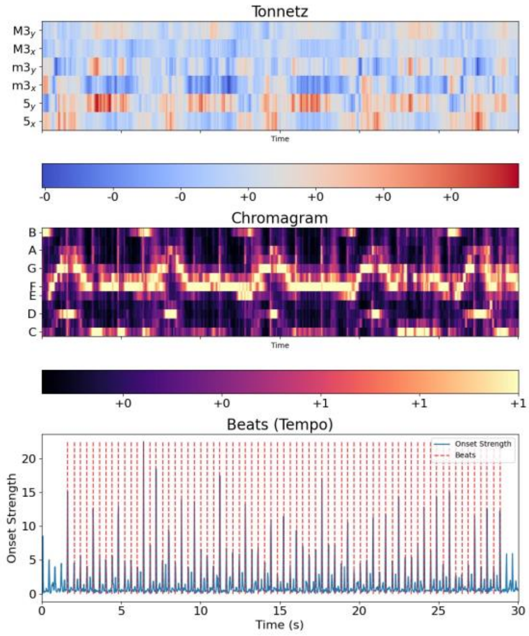


Fig. 2. Tonnetz spectrogram, chromagram and tempo of a rock track.

handle nonlinear problems. Using the base features from Table 3, the best hyperparameters found were a value of 10 for C and 0.01 for gamma.

b) KNN: The best values for n_neighbors, weights, and metric were selected through grid search. Using the base features from Table 3, the best hyperparameters found were 7 for n_neighbors, 'distance' for weights, and 'euclidean' for metric.

c) XGBoost: In this model, the RandomizedSearchCV function from the xgboost library was used to find the best hyperparameters n_estimators, max_depth, learning_rate, subsample, and colsample_bytree. Using the base features from Table 3, the best hyperparameters found were 200 for n_estimators, 6 for max_depth, 0.1 for learning_rate, 0.8 for subsample, and 0.7 for colsample_bytree.

We also conducted experiments integrating Deng's active learning method [6] during the training of each model. We chose Deng's active learning method because of its proven effectiveness in various domains, including text and image classification. Its ability to iteratively improve model performance by selecting the most informative samples aligns well with our goal of enhancing music genre classification accuracy.

For this, an active learning function was used, which iteratively selects the samples with the highest uncertainty from the pool dataset to add them to the training set. In each iteration, the model is trained with the expanded training set and its performance is evaluated. This process is repeated until the defined number of iterations is completed.

The active learning method allows the model to improve its performance by incorporating informative samples in a controlled manner. A block diagram of the method is shown in Figure 3.

We used a value of 10 for the parameter m, which represents the number of iterations in the active learning process. This value was chosen based on preliminary experiments that balanced computational efficiency and performance improvement. In each iteration, the model is trained with the expanded training set, incorporating samples from the pool set with the highest uncertainty. We divided the data into 20% for the initial set and 80% for the pool set.

After training and evaluating the models in their traditional form and also by adapting the active learning method, using the base features from Table 3, we used 30% of the total data for the traditional approach and 30% of the pool set for the active learning method. The results obtained are shown in the second column of Table 5. These show that the use of active learning significantly improves the accuracy of the models.

In the case of SVM, accuracy increases from 66% to 75%, representing a considerable improvement. Similarly, KNN improves from 60% to 69.7%, and XGBoost from 63.67% to 74%. These improvements suggest that active learning allows the models to focus on the most informative samples, resulting in better generalization and performance.

In comparison, traditional models without active learning present lower accuracies, indicating the effectiveness of Deng's active learning approach [6] in MGC problem. However, as mentioned in the introduction of this paper, we are also interested in demonstrating that the musical feature tonnetz influences the results. For this reason, a series of experiments were conducted to evaluate the effect of independently adding each musical feature: tonnetz, chroma, and tempo. The results of these experiments are also presented in Table 5, spanning the third to fifth columns.

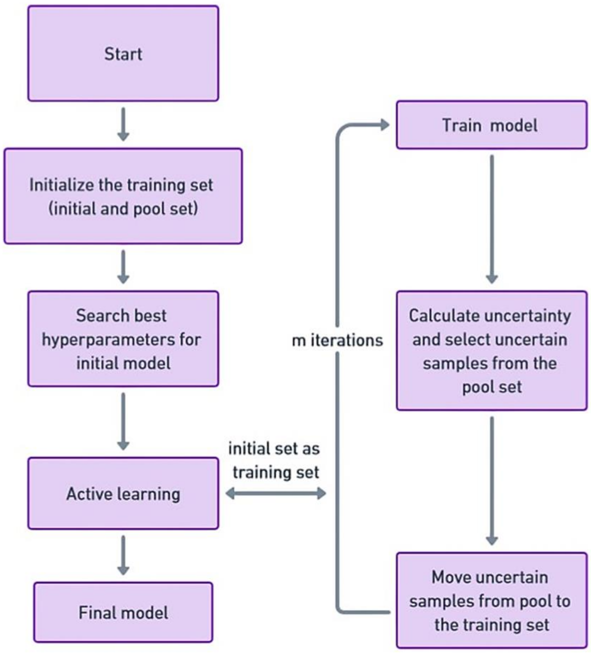

Start
Initialize the training set (initial and pool set)
Search best hyperparameters for initial model
Active learning
Final model
initial set as training set
m iterations
Train model
Calculate uncertainty and select uncertain samples from the pool set
Move uncertain samples from pool to the training set

Fig. 3. Block diagram of active learning.

Reviewing the results of the effect of independently adding each musical feature, it is noticeable that the accuracy of the SVM model improves significantly by adding the tonnetz feature, reaching 73.3%, and with active learning, this accuracy increases to 78.1%. For the KNN model, the addition of chroma shows a significant improvement, especially with active learning, where accuracy reaches 73%. In the case of XGBoost, the inclusion of tonnetz increases accuracy to 66%, and with active learning, it rises to 75%. These results highlight the importance of tonal and rhythmic features in music genre classification, particularly emphasizing the influence of tonnetz.

After verifying our first hypothesis, we conducted a series of comprehensive experiments by combining the musical features to maximize performance in each model. The best and only combination reported was Tonnetz + Chroma for SVM active. The result of this experiment is presented in the sixth and last column of Table 5.

The results show that combining features can lead to significant improvements in model accuracy. The SVM active model achieved the highest accuracy with a combination of tonnetz and chroma, reaching 80.3%. Its confusion matrix is shown in Figure 4.

These results highlight that combining musical features can be beneficial for optimizing model performance. However, there remains a noticeable trend in the relevance of the tonnetz feature over chroma and tempo.

For the statistical analysis, techniques such as paired t-tests could be conducted to compare the performance differences between traditional and active learning methods.

Table 5. Accuracy of our different models.

Model	**Base features (%)**	**Adding tonnetz (%)**	**Adding chroma (%)**	**Adding tempo (%)**	**Adding tonnetz and chroma (%)**
SVM	66	**73.3**	66.6	68	-
KNN	60	**69**	65.1	61.2	-
XGBoost	63.67	**66**	64.2	63.6	-
SVM active	75	78.1	77.2	76.3	**80.3**
KNN active	69.7	66.3	**73**	67.8	-
XGBoost active	74	**75**	73	72.3	-

The results would likely indicate that the performance improvements are statistically significant, reinforcing the effectiveness of the active learning approach.

As a final comparison, Table 6 includes the best performance obtained from the experiments in this work compared to state-of-the-art works that have evaluated their models with the GTZAN dataset.

Table 6 shows that our proposed model, SVM with active learning, achieved an accuracy of 80.3%, surpassing the results of Ghildiyal et al. [8] and Elbir et al. [3], who reached 74.3% with Decision Tree and 72.7% with SVM, respectively. Although Qi et al.'s [7] KNN model achieved an accuracy of 90%, significantly higher than the other models, as previously mentioned, this high performance is likely influenced by their use of nearly 18,000 additional samples from the Spotify developer API during training, beyond the GTZAN dataset. These results demonstrate the effectiveness of our approach and highlight the relevance of active learning and musical features, particularly tonnetz, in music genre classification.

4 Conclusion and Future Work

The results of this study highlight the significant impact of active learning and tonal features on the accuracy of music genre classification models. The active learning approach proved to be highly effective, allowing the models to focus on the most informative samples, which in turn resulted in better generalization and overall performance. Specifically, the SVM model with active learning achieved an accuracy of 80.3%, surpassing traditional models and showcasing the efficacy of integrating active learning with carefully selected musical features.

Our experiments demonstrated that the tonnetz feature plays a crucial role in enhancing model performance, particularly when combined with chroma features. This finding underscores the importance of incorporating harmonic, tonal, and rhythmic features to improve the classification accuracy of ML models for MGR. The results suggest that these features capture essential aspects of musical structure that are not fully leveraged by traditional spectral features alone.

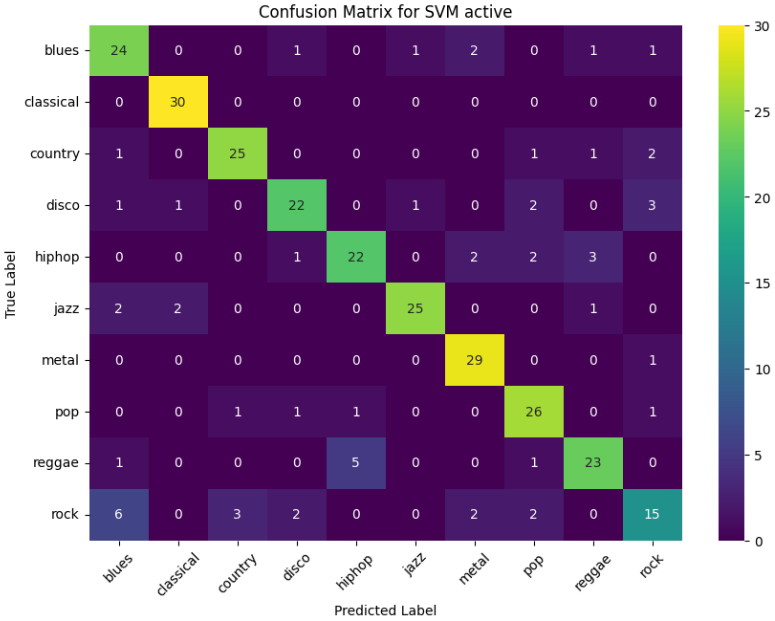


Fig. 4. GTZAN dataset confusion matrix for SVM active using Tonnetz and chroma.

Table 6. Comparative analysis with models of state of art.

Reference	Model	Accuracy %
Qi et al. [7]	KNN	90
This Work	SVM active	80.3
Ghildiyal et al. [8]	Decision Tree	74.3
Elbir et al. [3]	SVM	72.7

However, while the results are promising, the study's scalability to larger and more diverse datasets needs further exploration. The GTZAN dataset, though widely used, is relatively small and homogeneous. To ensure the robustness and generalizability of the proposed approach, it will be necessary to validate the models on larger datasets that reflect the diversity of real-world music streaming platforms. Incorporating additional samples from such platforms, as done by Qi et al. [7], could help address this limitation and provide more comprehensive insights into the model's performance in varied musical contexts.

For future work, we plan to expand the dataset by incorporating additional samples from the developer API of a music streaming platform, as mentioned earlier. This expansion will allow us to test the scalability and robustness of our approach on a broader range of genres and musical styles. Additionally, we aim to explore alternative feature representations, such as auditory filter banks proposed by the authors in [10], to evaluate their effectiveness in MGR. Another avenue for future research is the exploration of convolutional neural network (CNN) models, similar to those employed by Ghildiyal et al. [8], to assess their potential in further improving genre classification accuracy.

By addressing these areas, we hope to develop more robust and accurate models for music genre classification, leveraging advanced feature extraction techniques and larger, more diverse datasets. Such efforts will contribute to advancing the field of MGR and improving the practical applications of genre classification in the music industry.

References

1. Pachet, F., Cazaly, D.: A classification of musical genre. In: Proceedings of RIAO Content-Based Multimedia Information Access Conference (2000)
2. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on speech and audio processing, vol. 10, no. 5, pp. 293–302 (2002) doi:10.1109/TSA.2002.800560
3. Elbir, A., Çam, H. B., İyican, M. E., Öztürk, B., Aydın, N.: Music genre classification and recommendation by using ML techniques. In: 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), IEEE, pp. 1–5 (2018)
4. Korzeniowski, F., Widmer, G.: Feature learning for chord recognition: The deep chroma extractor. In: Proceedings of the 17th International Society for Music Information Retrieval Conference, p. 37–43 (2016) doi: 10.48550/arXiv.1612.05065
5. Humphrey, E. J., Cho, T., Bello, J. P.: Learning a robust tonnetz-space transform for automatic chord recognition. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 453–456 (2012) doi: 10.1109/ICASSP.2012.6287914.
6. Deng, G., Ko, Y. C.: Active learning music genre classification based on support vector machine. Advances in Multimedia, vol. 2022, no. 1, p. 4705272 (2022) doi: 10.1155/2022/4705272
7. Qi, Z., Rahouti, M., Jasim, M. A., Siasi, N.: Music genre classification and feature comparison using ml. In: Proceedings of the 2022 7th International Conference on Machine Learning Technologies, pp. 42–50 (2022) doi: 10.1145/3529399.352940
8. Ghildiyal, A., Singh, K., Sharma, S.: Music genre classification using ML. In: Fourth International Conference on Electronics, Communication and Aerospace Technology IEEE, pp. 1368–1372 (2020)
9. Sturm, B. L.: An analysis of the GTZAN music genre dataset. In: Proceedings of the Second International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies, pp. 7–12 (2012) doi: 10.1145/2390848.239085
10. Velazquez-Lopez, O., Oropeza-Rodriguez, J. L., Suarez-Guerra, S.: Application of auditory filter-banks in polyphonic music transcription. Computación y Sistemas, vol. 26, no. 4, pp. 1421–1428 (2022) doi:10.13053/CyS-26-4-4271

ISSN 1870-4069

LLAMA Assisted Nutritional Recipe Suggestions: Integrating the Dietary Quality Index for Health Conscious Cooking

Diego Estrada-Beltrán, Miguel Gonzalez-Mendoza,
Raul Monroy-Borja, Gilberto Ochoa-Ruiz,
Janet Gutiérrez-Uribe, Astrid Domínguez-Uscanga

Tecnológico de Monterrey,
Escuela de Ingeniería y Ciencias,
Mexico

`{a01252679, mgonza, raulm, gilberto.ochoa, jagu, astrid.dominguez}@tec.mx`

Abstract. Choosing what to eat daily can be challenging due to the variety of ingredients and their differing nutritional values, often leading to health problems such as cardiovascular diseases and psychological disorders. The recent surge in popularity of Large Language Models (LLMs) like ChatGPT, Gemini, and Llama has made it easier for users to obtain information and recommendations from various fields more quickly than traditional online searches. In this study, we trained Llama models using datasets on ingredients and their nutritional values, fast food menus with nutritional values, and the Dietary Quality Index (DQI) to provide personalized recipe suggestions with accurate nutritional information. We compare the performance of Retrieval Augmented Generation (RAG) with Llama-2 model, Llama-3, and the default ChatGPT model in terms of recipe accuracy, nutritional value precision, and user-friendliness. Our findings aim to demonstrate the potential of LLMs in improving dietary health through accessible and understandable nutritional guidance, while also addressing the common problem of hallucination in LLMs by using RAG for context information.

Keywords: DQI, Llama, LLM, RAG, nutritional daily intake, recipe recommendation.

1 Introduction

In kitchens around the globe, a daily dilemma unfolds: “What should I cook today?” With the abundance of recipes and ingredients making a single meal becomes a difficult task, and, beyond this problem, nutritional values of the food are another problem to take in account. With the lack of nutritional knowledge and the lack of knowledge to interpret the information provided by nutrition labels for most people [2, 4], this dilemma becomes hard to answer. In 1990, the Nutrition Labeling and Education Act (NLEA) was introduced in the United States, marking a pivotal moment in public health policy.

Its primary objective was to give consumers the necessary tools to make informed dietary choices. By using standardized labels with essential information, people could opt for healthier products. However, despite the widespread adoption of nutritional labels, research indicates that a substantial portion of the population overlooks or struggles to interpret these labels, Campos et al. (2011) showed that nutrition labels that require calculations concerning nutrient amounts and serving sizes are confusing to many consumers, particularly those with lower education and literacy skills so, a graphical view would be more helpful. Grunert et al. (2010) highlighted that from a questionnaire of 921 people, 27% checked the nutritional labels and from this, 70 to 90% could correctly interpret the information.

This lack of nutritional understanding can lead to imbalanced dietary patterns [7], characterized by both excessive and deficient consumption of certain nutrients. Such dietary imbalances are associated with a spectrum of non-communicable diseases (NCDs), encompassing immediate complications and long-term health consequences, including cardiovascular diseases, diabetes, hypertension, stroke, cancer, dental caries, asthma, and various psychological disorders such as depression [9, 11].

Alarmingly, the International Diabetes Federation reports that approximately 415 million people worldwide suffer from diabetes, with incidence rates projected to surge by over 50% by 2040. Furthermore, the Global Burden of Disease Study underscores the significant contribution of dietary factors to levels of malnutrition, obesity, and overweight, with 11 million preventable premature deaths annually attributed to unreasonable dietary habits [10].

The advancements in machine learning have spurred the development of numerous recommendation systems (RSs) aimed at addressing challenges in recipe suggestions and nutritional knowledge. For instance, Zhang et al (2022) introduced the MaOO model, a vector optimization algorithm designed to optimize multiple objective functions [14]. By seamlessly integrating ingredients' nutritional value, food diversity, and user dietary patterns, the MaOO model correctly suggests suitable food options for users. Similarly, Teng et al. (2012) proposed a methodology leveraging pointwise mutual information (PMI) coupled with support vector machines (SVMs).

This approach establishes correlations among ingredients found in diverse recipes, identifying the most frequently used ingredients. By employing SVMs, it helped to further refine recipe suggestions [12]. or the most akin to this work, introduced by Aljbawi Bushra (2020) who utilized GPT-2 to generate recipe suggestions based on input prompts containing ingredients by training it with different recipes and the ingredients used [1].

The introduction of OpenAI's GPT-3 public model in November 2022 has sparked a surge in interest surrounding Generative Artificial Intelligence (GAI) models. With reports indicating that over 100 million individuals engaged with the model in January 2023 alone [3], GAI technologies have become a focal point of contemporary discourse. Utilizing Generative Pre-trained Transformers (GPT), these models are trained on extensive textual datasets, empowering them to produce coherent and contextually relevant responses. Notably, the GPT-3 model boasts an impressive 175 billion parameters, derived from textual data converted into tokens.

Subsequently, based on user prompts, a retriever mechanism identifies and retrieves the most pertinent information to address user inquiries [8]. As a result of the widespread acclaim garnered by the GPT model, numerous other companies have ventured into the development of their own Generative Artificial Intelligence (GAI) models. Among these is Meta's Llama-2 model, introduced in July 2023 [13]. A refined iteration of the original Llama 1 model, Llama-2 has been trained on extensive datasets, resulting in versions with parameters ranging from 7 billion to 70 billion.

On April 18, 2024, Meta released a new model named Llama-3, with parameters ranging from 8 to 70 billion. It was trained on 15 trillion tokens from public sources, making its training dataset seven times larger than that of Llama-2. Additionally, Llama-3 utilizes a tokenizer with 128,000 tokens to encode language more efficiently. Meta also modified the code to enhance the model's manipulability [6].

Capitalizing on the growing popularity of these models and recognizing the imperative to address malnutrition risks, we have developed an innovative model with enhanced user interaction. This model not only provides recipe suggestions based on available ingredients but also offers personalized nutritional guidance tailored to individual needs. Equipped with memory functionality, our model dynamically adapts to users' dietary preferences, leveraging Dietary Quality Index (DQI) scores to monitor and adjust nutritional intake. This adaptive approach ensures the delivery of healthy recipe suggestions suitable for individuals of all ages and varying culinary skills.

Building upon these capabilities, our model goes a step further by offering recipe suggestions based on portion sizes, enabling more precise monitoring of nutritional intake. This enhanced feature ensures a more controlled approach to recipe recommendations, facilitating accurate tracking of nutrition and promoting healthier dietary habits.

2 Materials and Methods

2.1 Data Collection

We collected data through web scraping from two primary sources: FoodData Central USDA and MenuWithNutrition. From FoodData Central USDA, we obtained a dataset comprising ingredients along with their corresponding nutritional values. The MenuWithNutrition website provided us with a diverse dataset of USA fast food menus, including detailed nutritional information for each menu item. Additionally, to ensure that our model recommends recipes aligned with nutritional guidelines, we integrated data from the Dietary Quality Index (DQI) document.

This document informed our model to prioritize recipes that contribute to achieving the nutritional scores outlined in the DQI. For the data extraction process, we utilized the request library to access the respective websites. Specifically, urllib.request and urlopen were employed to fetch data from the URLs of these websites. Subsequently, BeautifulSoup was utilized to parse the HTML structure and extract the relevant information needed from the web pages.

Table 1. Fraction of USDA dataset.

Main food description	Food code	WWEIA Number	WWEIA Description	Energy (kcal)	Protein (g)
Milk, human	11000000	9602	Human milk	70	1.03
Milk, NFS	11100000	1004	Milk, reduced fat	52	3.33
Milk, whole	11111000	1002	Milk, whole	61	3.27
Milk, reduced fat (2%)	11112110	1004	Milk, reduced fat	50	3.36
Milk, low fat (1%)	11112210	1006	Milk, lowfat	43	3.38

2.2 Data Understanding

FoodData Central USDA. FoodData Central (FDC) is a comprehensive food and nutrient database maintained by the United States Department of Agriculture (USDA), offering detailed information on the nutritional composition of a wide array of foods consumed in the United States. This database encompasses essential data on nutrients such as vitamins, minerals, carbohydrates, proteins, fats, and other components present in foods. From FDC, we utilized the Food and Nutrient Database for Dietary Studies 2019-2020 (FNDDS 2019-2020). This specialized database provides nutritional details for foods and beverages reported in ”What We Eat in America,” a component of the National Health and Nutrition Examination Survey (NHANES). This table, with 5625 rows and 69 columns, contains food descriptions and their respective food codes from the FNDDS.

Each 8-digit food code starts with a digit representing one of nine major food groups: Milk and Milk Products, Meat, Poultry, Fish, and Mixtures, Eggs, Dry Beans, Peas, Other Legumes, Nuts, and Seeds, Grain Products, Fruits, Vegetables, Fats, Oils, and Salad Dressings, Sugars, Sweets, and Beverages. The remaining digits specify subgroups, with codes ranging from 11000000 to 99998210. It includes WWEIA Category numbers and descriptions for 170 subgroups, ranging from 1002 to 9999, which classify foods into specific subgroups. The rest of the columns provide nutritional values (micro and macronutrients) per 100 grams of each food item.

MenuWithNutrition. MenuWithNutrition is a website dedicated to providing detailed information on menus from various fast food chains across the USA. They compile data from multiple sources and meticulously verify the accuracy of nutritional values. The website is designed to be comprehensive and user-friendly, helping individuals understand the nutritional content of food items from different restaurant chains. From MenuWithNutrition, we obtained a database similar to that of the USDA. This database includes information on restaurants, menu items, and nutritional values such as macronutrients (fats, proteins, carbohydrates, fiber) and micronutrients (monounsaturated fats, polyunsaturated fats, among others).

Dietary Quality Index. The Dietary Quality Index (DQI) is a metric developed by Soowon Kim et al. [5] to evaluate the overall quality of daily food intake. It is structured into four key categories aimed at assessing different nutritional aspects:

1. **Variety:** This category evaluates the diversity of food sources within the diet, including proteins from various food groups such as meats, fruits, vegetables, dairy, and grains.

Table 2. Fraction of MenuWithNutrition dataset.

Restaurant Name	Food Name	Total Fat	Cholesterol	Sodium	Total Carbohydrate	Protein
aandw restaurant	chocolate cone	4.5 g	15 mg	105 g	26 g	3 g
aandw restaurant	root beer float	5.2 g	39 mg	104 g	70.4 g	2.1 g
aandw restaurant	diet root beer float	5.2 g	39 mg	104 g	31.1 g	2.1 g
aandw restaurant	root beer freeze	18 g	70 mg	400 g	150 g	16 g
aandw restaurant	chocolate shake	28.8 g	124 mg	200 g	100 g	11.2 g

2. **Adequacy:** Adequacy assesses whether the intake of essential dietary elements meets recommended levels to ensure a healthy diet and prevent undernutrition.
3. **Moderation:** This examines the intake of foods and nutrients associated with chronic diseases, emphasizing moderation of total fats, saturated fats, cholesterol, and sodium.
4. **Overall Balance:** The final category evaluates the overall balance of the diet in terms of energy sources and fatty acid composition, aiming for proportional intake across these categories.

2.3 Llama and Retrieval Augmented Generation

The Llama-2 and Llama-3 models, with 7 billion and 8 billion parameters respectively, were used as the foundation for our approach. These models, developed by Meta, were downloaded from Hugging Face. To fine-tune and manipulate these models, we utilized the Langchain library, known for its versatile capabilities in model adaptation. An essential component of our method was the system prompt template, which guided the model's responses. The template used was as follows:

- “”Use the following pieces of information to answer the user's question. If you don't know the answer, just say that you don't know, don't try to make up an answer.
- Context: context.
- Question: question.

You task is to suggest recipes and give people information about nutritional information about their food based on the Dietary Quality Index (DQI). The necessary nutritional values to take into account for the DQI and the recipe suggestions are: Protein (g), Lipids (g), Fiber (g), Ascorbic Acid (mg), Cholesterol (mg), Saturated Fatty Acids (g), Calcium (mg), Iron (mg), Sodium (mg), Carbohydrates (g), SFAs, MFAs, PUFAs and Total Energy (Kcal).

- Helpful Answer: ””

This template ensures the model answers questions only if it knows the answer. Also, since it is a safe mode, it doesn't respond to harmful queries, thus eliminating the need for additional safety instructions in the template.

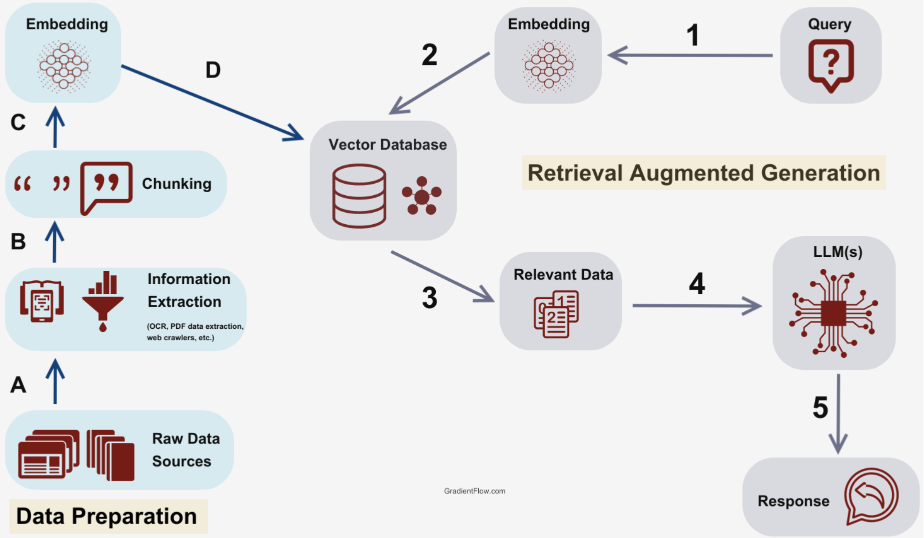


Fig. 1. Retrieval-augmented generation.

By including the context and question parts, we enable the model to retrieve relevant information from the uploaded data to answer questions accurately. The task specified in the template directs the model to suggest recipes and provide their nutritional information, aligning the model's responses with our objectives. To prepare our data for the model, we performed ETL (Extract, Transform, Load) operations. Initially, we extracted comprehensive data through web scraping. Next, we cleaned the data by removing null and duplicate values to ensure data integrity and consistency. Finally, we loaded the cleaned data into the model.

The cleaned datasets were then formatted into .csv files, facilitating seamless integration with the Langchain RAG technique. The RAG (Retrieval-Augmented Generation) technique served as a pivotal component in augmenting the Llama models' knowledge with additional data. This technique comprises five key steps: Load, Split, Store, Retrieve, and Generate. Through this process, we were able to effectively incorporate diverse datasets into the model, enriching its understanding of nutritional information and enhancing its capacity to generate relevant recipe suggestions based on user queries:

1. **Load:** The initial step of the RAG technique involves loading the three datasets we possess. Langchain, our chosen tool for this task, is capable of accepting various document formats such as PDFs, CSV files, SQL databases, or even website text in HTML format.

2. **Split:** Given that LLM models have constraints regarding the length of input they can process, we implement a splitting process for our documents. Each document is divided into manageable chunks, with each chunk containing approximately 1000 characters. To maintain continuity and coherence, these chunks overlap by 200

characters from one document to the next. This approach ensures that the model can follow the sequential order of the documents while processing them.

3. **Store:** Following the splitting process, we transform these chunks into embeddings, which are numerical representations of the text data. These embeddings are then stored in a vector database, which we opted to utilize the Chroma vector database for this purpose. Storing the embeddings in a vector database facilitates efficient retrieval based on similarity search, a crucial aspect of the subsequent steps.

4. **Retrieve:** Using the stored embeddings in the vector database, we perform similarity searches to retrieve the most relevant embeddings corresponding to the prompt provided by the user. This retrieval process ensures that the model accesses pertinent information from the datasets that aligns closely with the user's query, enhancing the relevance and accuracy of the model's responses.

5. **Generate:** The final step of the RAG technique involves the generation of responses by the model. This process unfolds as a chain, beginning with the user's prompt, followed by the retrieval of relevant information from the stored embeddings. The model then constructs a response based on this retrieved information, synthesizing it with the user's prompt to generate a coherent and contextually relevant output.

3 Results

To evaluate the models' performance, we used the following questions to assess their ability to generate creative and accurate responses, verify the accuracy of nutritional values, and correctly retrieve data for contextual understanding:

1. Retrieve nutritional information for a food item (e.g., chicken wing).
2. Provide a recipe that meets the necessary nutritional intake according to the Dietary Quality Index (DQI).
3. Explain the differences between fast food and homemade food.
4. Create a recipe centered around a specific ingredient.
5. Compare foods to identify healthier meal options, such as suggesting substitutes for mayonnaise in sandwiches.

3.1 Llama-2 7b Answers

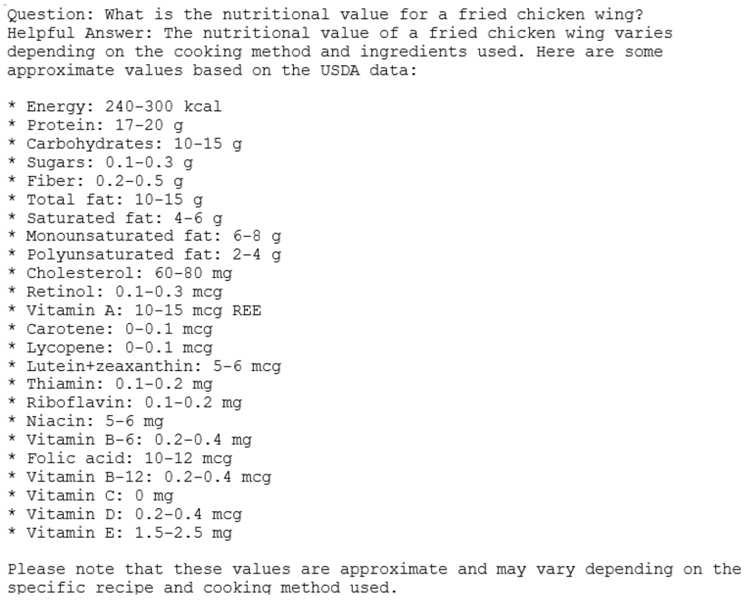
Shown from Figure 2 to Figure 6

3.2 Llama-3 8b Answers

Shown from Figure 7 to Figure 11

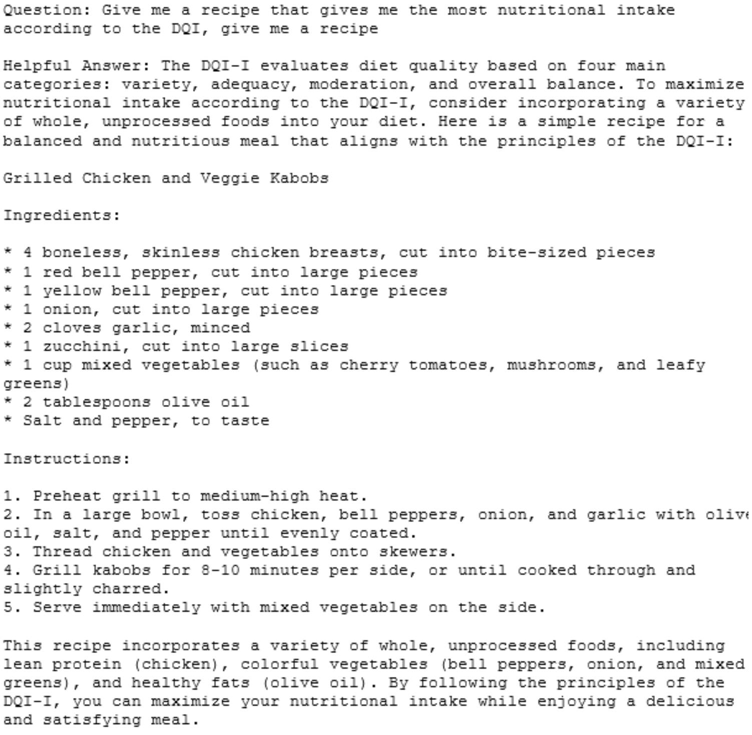
3.3 Default ChatGPT-3 Answers

Shown from Figure 12 to Figure 16



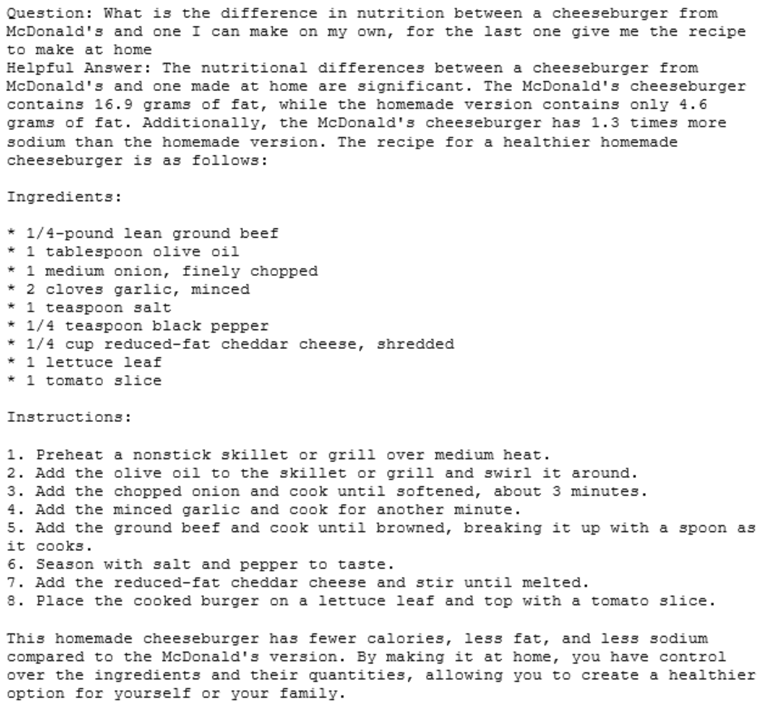
```
Question: What is the nutritional value for a fried chicken wing?
Helpful Answer: The nutritional value of a fried chicken wing varies
depending on the cooking method and ingredients used. Here are some
approximate values based on the USDA data:

* Energy: 240-300 kcal
* Protein: 17-20 g
* Carbohydrates: 10-15 g
* Sugars: 0.1-0.3 g
* Fiber: 0.2-0.5 g
* Total fat: 10-15 g
* Saturated fat: 4-6 g
* Monounsaturated fat: 6-8 g
* Polyunsaturated fat: 2-4 g
* Cholesterol: 60-80 mg
* Retinol: 0.1-0.3 mcg
* Vitamin A: 10-15 mcg REE
* Carotene: 0-0.1 mcg
* Lycopene: 0-0.1 mcg
* Lutein+zeaxanthin: 5-6 mcg
* Thiamin: 0.1-0.2 mg
* Riboflavin: 0.1-0.2 mg
* Niacin: 5-6 mg
* Vitamin B-6: 0.2-0.4 mg
* Folic acid: 10-12 mcg
* Vitamin B-12: 0.2-0.4 mcg
* Vitamin C: 0 mg
* Vitamin D: 0.2-0.4 mcg
* Vitamin E: 1.5-2.5 mg

Please note that these values are approximate and may vary depending on the
specific recipe and cooking method used.
```

Fig. 2. Question 1 Llama 2.



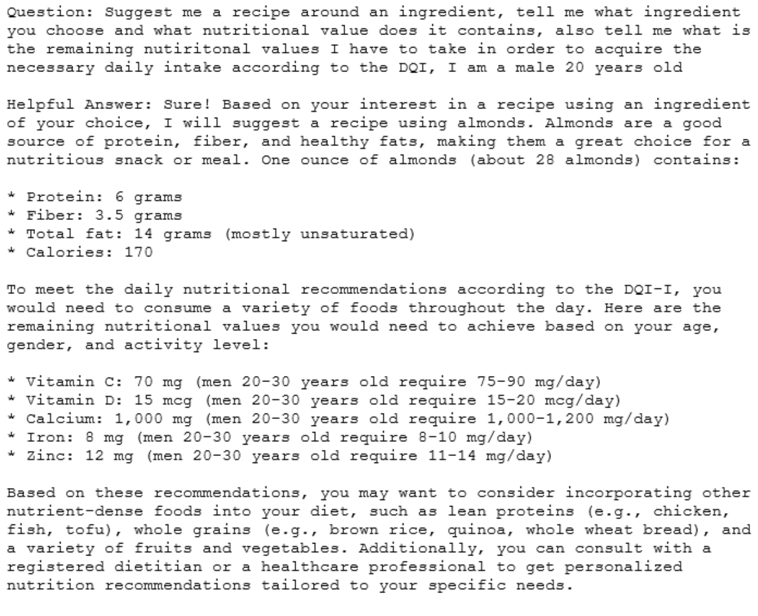
```
Question: Give me a recipe that gives me the most nutritional intake
according to the DQI, give me a recipe

Helpful Answer: The DQI-I evaluates diet quality based on four main
categories: variety, adequacy, moderation, and overall balance. To maximize
nutritional intake according to the DQI-I, consider incorporating a variety
of whole, unprocessed foods into your diet. Here is a simple recipe for a
balanced and nutritious meal that aligns with the principles of the DQI-I:

Grilled Chicken and Veggie Kabobs

Ingredients:

* 4 boneless, skinless chicken breasts, cut into bite-sized pieces
* 1 red bell pepper, cut into large pieces
* 1 yellow bell pepper, cut into large pieces
* 1 onion, cut into large pieces
* 2 cloves garlic, minced
* 1 zucchini, cut into large slices
* 1 cup mixed vegetables (such as cherry tomatoes, mushrooms, and leafy
greens)
* 2 tablespoons olive oil
* Salt and pepper, to taste

Instructions:

1. Preheat grill to medium-high heat.
2. In a large bowl, toss chicken, bell peppers, onion, and garlic with oliv
oil, salt, and pepper until evenly coated.
3. Thread chicken and vegetables onto skewers.
4. Grill kabobs for 8-10 minutes per side, or until cooked through and
slightly charred.
5. Serve immediately with mixed vegetables on the side.

This recipe incorporates a variety of whole, unprocessed foods, including
lean protein (chicken), colorful vegetables (bell peppers, onion, and mixed
greens), and healthy fats (olive oil). By following the principles of the
DQI-I, you can maximize your nutritional intake while enjoying a delicious
and satisfying meal.
```

Fig. 3. Question 2 Llama 2.



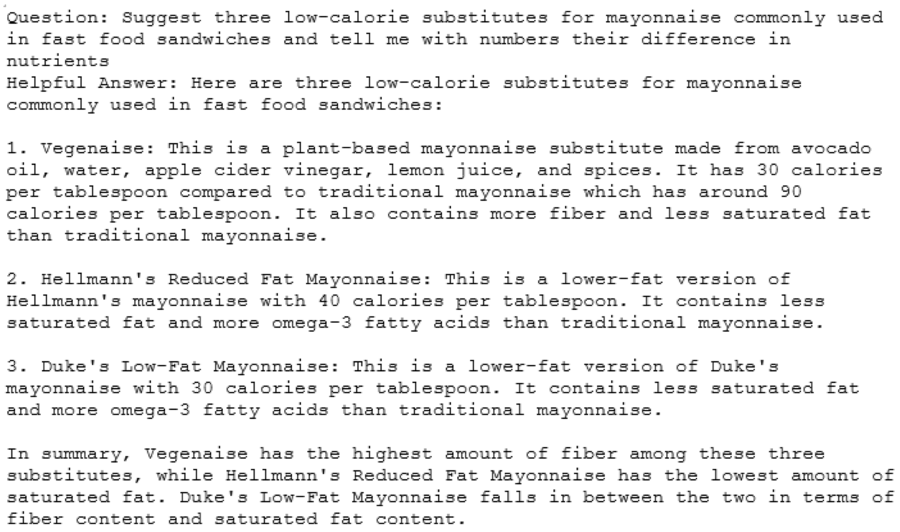
```
Question: What is the difference in nutrition between a cheeseburger from
McDonald's and one I can make on my own, for the last one give me the recipe
to make at home
Helpful Answer: The nutritional differences between a cheeseburger from
McDonald's and one made at home are significant. The McDonald's cheeseburger
contains 16.9 grams of fat, while the homemade version contains only 4.6
grams of fat. Additionally, the McDonald's cheeseburger has 1.3 times more
sodium than the homemade version. The recipe for a healthier homemade
cheeseburger is as follows:

Ingredients:

* 1/4-pound lean ground beef
* 1 tablespoon olive oil
* 1 medium onion, finely chopped
* 2 cloves garlic, minced
* 1 teaspoon salt
* 1/4 teaspoon black pepper
* 1/4 cup reduced-fat cheddar cheese, shredded
* 1 lettuce leaf
* 1 tomato slice

Instructions:

1. Preheat a nonstick skillet or grill over medium heat.
2. Add the olive oil to the skillet or grill and swirl it around.
3. Add the chopped onion and cook until softened, about 3 minutes.
4. Add the minced garlic and cook for another minute.
5. Add the ground beef and cook until browned, breaking it up with a spoon as
it cooks.
6. Season with salt and pepper to taste.
7. Add the reduced-fat cheddar cheese and stir until melted.
8. Place the cooked burger on a lettuce leaf and top with a tomato slice.

This homemade cheeseburger has fewer calories, less fat, and less sodium
compared to the McDonald's version. By making it at home, you have control
over the ingredients and their quantities, allowing you to create a healthier
option for yourself or your family.
```

Fig. 4. Question 3 Llama 2.



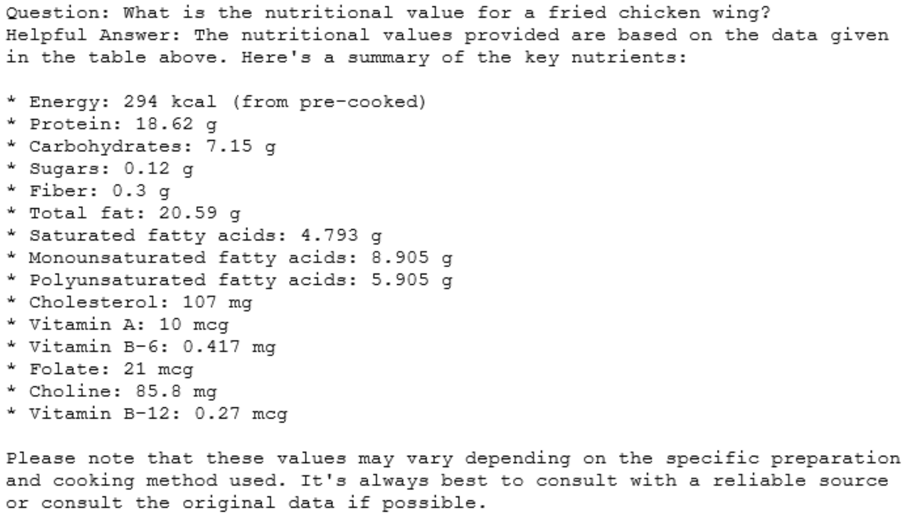
```
Question: Suggest me a recipe around an ingredient, tell me what ingredient
you choose and what nutritional value does it contains, also tell me what is
the remaining nutiritonal values I have to take in order to acquire the
necessary daily intake according to the DQI, I am a male 20 years old

Helpful Answer: Sure! Based on your interest in a recipe using an ingredient
of your choice, I will suggest a recipe using almonds. Almonds are a good
source of protein, fiber, and healthy fats, making them a great choice for a
nutritious snack or meal. One ounce of almonds (about 28 almonds) contains:

* Protein: 6 grams
* Fiber: 3.5 grams
* Total fat: 14 grams (mostly unsaturated)
* Calories: 170

To meet the daily nutritional recommendations according to the DQI-I, you
would need to consume a variety of foods throughout the day. Here are the
remaining nutritional values you would need to achieve based on your age,
gender, and activity level:

* Vitamin C: 70 mg (men 20-30 years old require 75-90 mg/day)
* Vitamin D: 15 mcg (men 20-30 years old require 15-20 mcg/day)
* Calcium: 1,000 mg (men 20-30 years old require 1,000-1,200 mg/day)
* Iron: 8 mg (men 20-30 years old require 8-10 mg/day)
* Zinc: 12 mg (men 20-30 years old require 11-14 mg/day)

Based on these recommendations, you may want to consider incorporating other
nutrient-dense foods into your diet, such as lean proteins (e.g., chicken,
fish, tofu), whole grains (e.g., brown rice, quinoa, whole wheat bread), and
a variety of fruits and vegetables. Additionally, you can consult with a
registered dietitian or a healthcare professional to get personalized
nutrition recommendations tailored to your specific needs.
```

Fig. 5. Question 4 Llama 2.



```
Question: Suggest three low-calorie substitutes for mayonnaise commonly used
in fast food sandwiches and tell me with numbers their difference in
nutrients
Helpful Answer: Here are three low-calorie substitutes for mayonnaise
commonly used in fast food sandwiches:

1. Vegenaise: This is a plant-based mayonnaise substitute made from avocado
oil, water, apple cider vinegar, lemon juice, and spices. It has 30 calories
per tablespoon compared to traditional mayonnaise which has around 90
calories per tablespoon. It also contains more fiber and less saturated fat
than traditional mayonnaise.

2. Hellmann's Reduced Fat Mayonnaise: This is a lower-fat version of
Hellmann's mayonnaise with 40 calories per tablespoon. It contains less
saturated fat and more omega-3 fatty acids than traditional mayonnaise.

3. Duke's Low-Fat Mayonnaise: This is a lower-fat version of Duke's
mayonnaise with 30 calories per tablespoon. It contains less saturated fat
and more omega-3 fatty acids than traditional mayonnaise.

In summary, Vegenaise has the highest amount of fiber among these three
substitutes, while Hellmann's Reduced Fat Mayonnaise has the lowest amount of
saturated fat. Duke's Low-Fat Mayonnaise falls in between the two in terms of
fiber content and saturated fat content.
```

Fig. 6. Question 5 Llama 2.

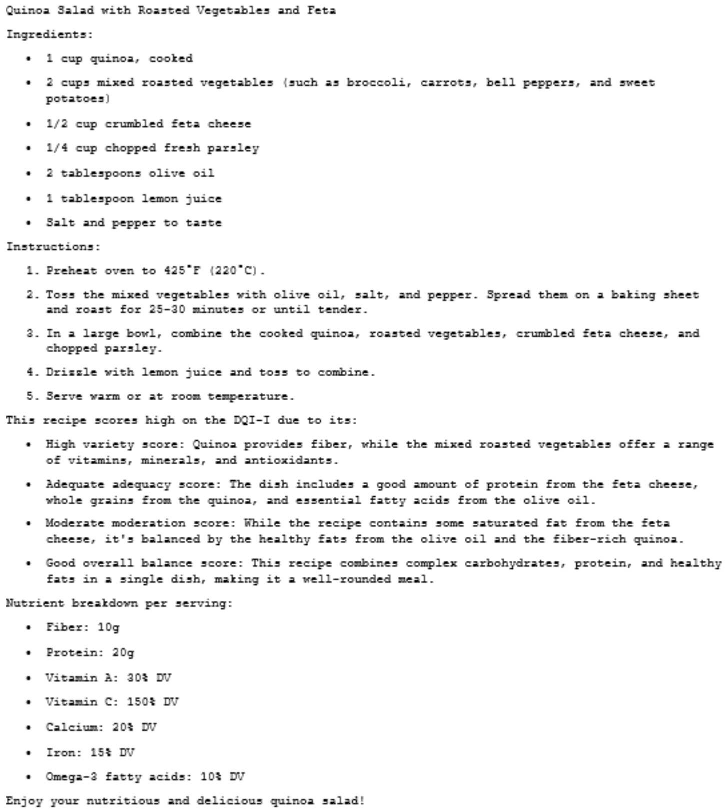


```
Question: What is the nutritional value for a fried chicken wing?
Helpful Answer: The nutritional values provided are based on the data given
in the table above. Here's a summary of the key nutrients:

* Energy: 294 kcal (from pre-cooked)
* Protein: 18.62 g
* Carbohydrates: 7.15 g
* Sugars: 0.12 g
* Fiber: 0.3 g
* Total fat: 20.59 g
* Saturated fatty acids: 4.793 g
* Monounsaturated fatty acids: 8.905 g
* Polyunsaturated fatty acids: 5.905 g
* Cholesterol: 107 mg
* Vitamin A: 10 mcg
* Vitamin B-6: 0.417 mg
* Folate: 21 mcg
* Choline: 85.8 mg
* Vitamin B-12: 0.27 mcg

Please note that these values may vary depending on the specific preparation
and cooking method used. It's always best to consult with a reliable source
or consult the original data if possible.
```

Fig. 7. Question 1 Llama 3.

4 Discussion

The models demonstrated varying capabilities in suggesting recipes and providing nutritional values. ChatGPT struggled to deliver accurate nutritional information, while Llama-2 opted for a range of nutritional values, acknowledging factors such as cooking methods, storage, and spices that influence nutritional content variability. In contrast, Llama-3 provided precise values by specifying exact ingredients, presenting more detailed and creative responses.



Quinoa Salad with Roasted Vegetables and Feta

Ingredients:

- **1 cup quinoa, cooked**
- **2 cups mixed roasted vegetables (such as broccoli, carrots, bell peppers, and sweet potatoes)**
- **1/2 cup crumbled feta cheese**
- **1/4 cup chopped fresh parsley**
- **2 tablespoons olive oil**
- **1 tablespoon lemon juice**
- **Salt and pepper to taste**

Instructions:

1. **Preheat oven to 425°F (220°C).**
2. **Toss the mixed vegetables with olive oil, salt, and pepper. Spread them on a baking sheet and roast for 25-30 minutes or until tender.**
3. **In a large bowl, combine the cooked quinoa, roasted vegetables, crumbled feta cheese, and chopped parsley.**
4. **Drizzle with lemon juice and toss to combine.**
5. **Serve warm or at room temperature.**

This recipe scores high on the DQI-I due to its:

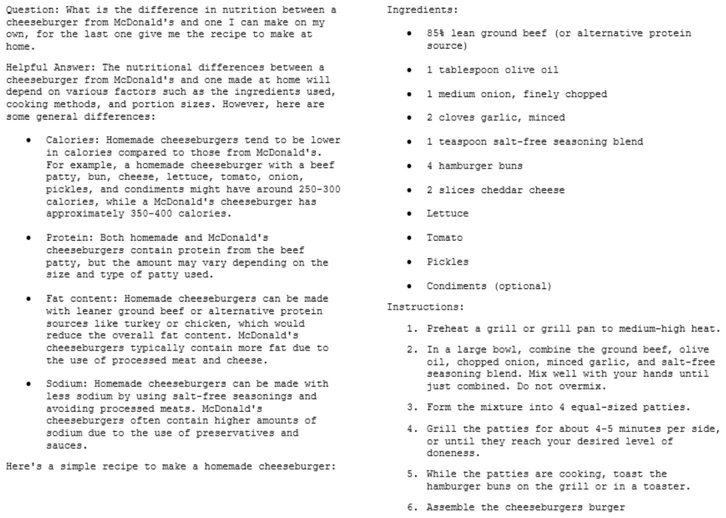
- **High variety score: Quinoa provides fiber, while the mixed roasted vegetables offer a range of vitamins, minerals, and antioxidants.**
- **Adequate adequacy score: The dish includes a good amount of protein from the feta cheese, whole grains from the quinoa, and essential fatty acids from the olive oil.**
- **Moderate moderation score: While the recipe contains some saturated fat from the feta cheese, it's balanced by the healthy fats from the olive oil and the fiber-rich quinoa.**
- **Good overall balance score: This recipe combines complex carbohydrates, protein, and healthy fats in a single dish, making it a well-rounded meal.**

Nutrient breakdown per serving:

- **Fiber: 10g**
- **Protein: 20g**
- **Vitamin A: 30% DV**
- **Vitamin C: 150% DV**
- **Calcium: 20% DV**
- **Iron: 15% DV**
- **Omega-3 fatty acids: 10% DV**

Enjoy your nutritious and delicious quinoa salad!

Fig. 8. Question 2 Llama 3.



Question: What is the difference in nutrition between a cheeseburger from McDonald's and one I can make on my own, for the last one give me the recipe to make at home.

Helpful Answer: The nutritional differences between a cheeseburger from McDonald's and one made at home will depend on various factors such as the ingredients used, cooking methods, and portion sizes. However, here are some general differences:

- Calories: Homemade cheeseburgers tend to be lower in calories compared to those from McDonald's. For example, a homemade cheeseburger with a beef patty, bun, cheese, lettuce, tomato, onion, pickles, and condiments might have around 250-300 calories, while a McDonald's cheeseburger has approximately 350-400 calories.
- Protein: Both homemade and McDonald's cheeseburgers contain protein from the beef patty, but the amount may vary depending on the size and type of patty used.
- Fat content: Homemade cheeseburgers can be made with leaner ground beef or alternative protein sources like turkey or chicken, which would reduce the overall fat content. McDonald's cheeseburgers typically contain more fat due to the use of processed meat and cheese.
- Sodium: Homemade cheeseburgers can be made with less sodium by using salt-free seasonings and avoiding processed meats. McDonald's cheeseburgers often contain higher amounts of sodium due to the use of preservatives and sauces.

Here's a simple recipe to make a homemade cheeseburger:

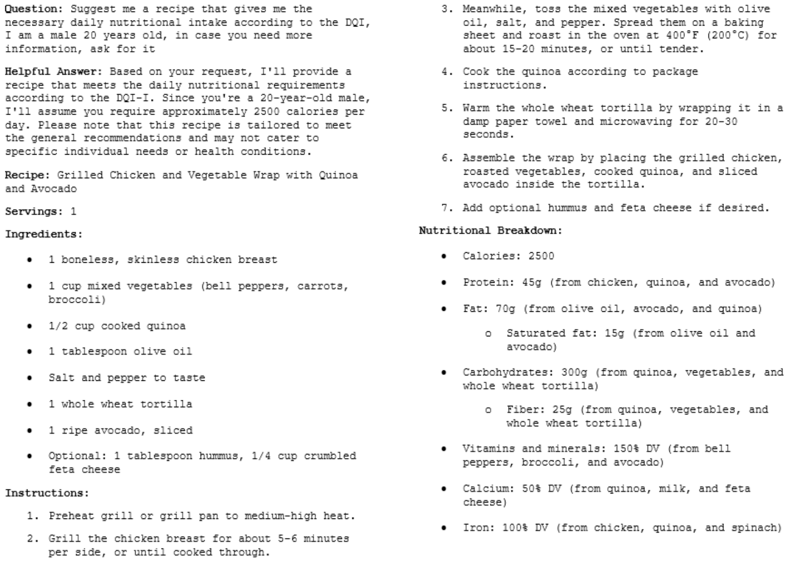
Ingredients:

- 85% lean ground beef (or alternative protein source)
- 1 tablespoon olive oil
- 1 medium onion, finely chopped
- 2 cloves garlic, minced
- 1 teaspoon salt-free seasoning blend
- 4 hamburger buns
- 2 slices cheddar cheese
- Lettuce
- Tomato
- Pickles
- Condiments (optional)

Instructions:

1. Preheat a grill or grill pan to medium-high heat.
2. In a large bowl, combine the ground beef, olive oil, chopped onion, minced garlic, and salt-free seasoning blend. Mix well with your hands until just combined. Do not overmix.
3. Form the mixture into 4 equal-sized patties.
4. Grill the patties for about 4-5 minutes per side, or until they reach your desired level of doneness.
5. While the patties are cooking, toast the hamburger buns on the grill or in a toaster.
6. Assemble the cheeseburgers burger

Fig. 9. Question 3 Llama 3.



Question: Suggest me a recipe that gives me the necessary daily nutritional intake according to the DQI, I am a male 20 years old, in case you need more information, ask for it

Helpful Answer: Based on your request, I'll provide a recipe that meets the daily nutritional requirements according to the DQI-I. Since you're a 20-year-old male, I'll assume you require approximately 2500 calories per day. Please note that this recipe is tailored to meet the general recommendations and may not cater to specific individual needs or health conditions.

Recipe: Grilled Chicken and Vegetable Wrap with Quinoa and Avocado

Servings: 1

Ingredients:

- 1 boneless, skinless chicken breast
- 1 cup mixed vegetables (bell peppers, carrots, broccoli)
- 1/2 cup cooked quinoa
- 1 tablespoon olive oil
- Salt and pepper to taste
- 1 whole wheat tortilla
- 1 ripe avocado, sliced
- Optional: 1 tablespoon hummus, 1/4 cup crumbled feta cheese

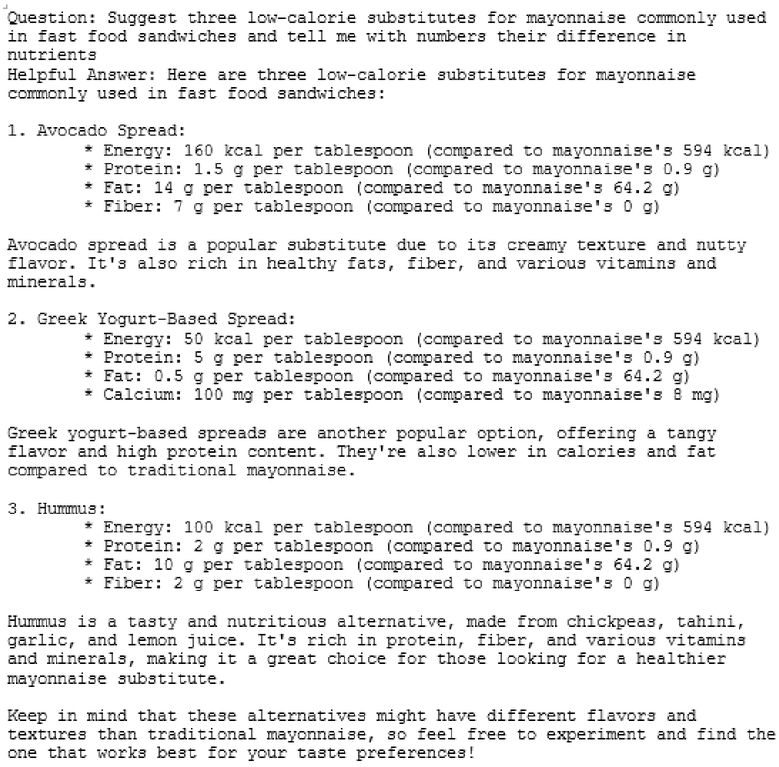
Instructions:

1. Preheat grill or grill pan to medium-high heat.
2. Grill the chicken breast for about 5-6 minutes per side, or until cooked through.
3. Meanwhile, toss the mixed vegetables with olive oil, salt, and pepper. Spread them on a baking sheet and roast in the oven at 400°F (200°C) for about 15-20 minutes, or until tender.
4. Cook the quinoa according to package instructions.
5. Warm the whole wheat tortilla by wrapping it in a damp paper towel and microwaving for 20-30 seconds.
6. Assemble the wrap by placing the grilled chicken, roasted vegetables, cooked quinoa, and sliced avocado inside the tortilla.
7. Add optional hummus and feta cheese if desired.

Nutritional Breakdown:

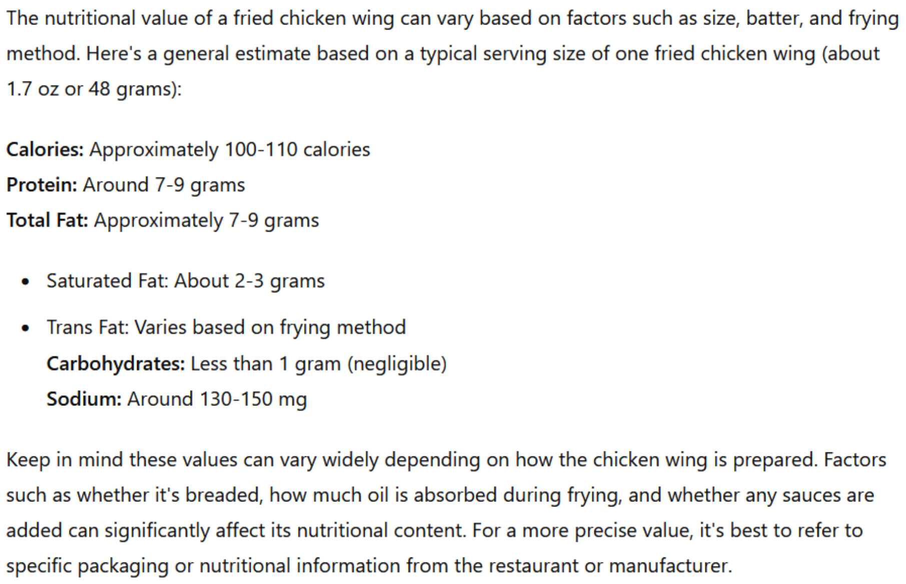
- Calories: 2500
- Protein: 45g (from chicken, quinoa, and avocado)
- Fat: 70g (from olive oil, avocado, and quinoa)
    - Saturated fat: 15g (from olive oil and avocado)
- Carbohydrates: 300g (from quinoa, vegetables, and whole wheat tortilla)
    - Fiber: 25g (from quinoa, vegetables, and whole wheat tortilla)
- Vitamins and minerals: 150% DV (from bell peppers, broccoli, and avocado)
- Calcium: 50% DV (from quinoa, milk, and feta cheese)
- Iron: 100% DV (from chicken, quinoa, and spinach)

Fig. 10. Question 4 Llama 3.



```
Question: Suggest three low-calorie substitutes for mayonnaise commonly used
in fast food sandwiches and tell me with numbers their difference in
nutrients
Helpful Answer: Here are three low-calorie substitutes for mayonnaise
commonly used in fast food sandwiches:

1. Avocado Spread:
        * Energy: 160 kcal per tablespoon (compared to mayonnaise's 594 kcal)
        * Protein: 1.5 g per tablespoon (compared to mayonnaise's 0.9 g)
        * Fat: 14 g per tablespoon (compared to mayonnaise's 64.2 g)
        * Fiber: 7 g per tablespoon (compared to mayonnaise's 0 g)

Avocado spread is a popular substitute due to its creamy texture and nutty
flavor. It's also rich in healthy fats, fiber, and various vitamins and
minerals.

2. Greek Yogurt-Based Spread:
        * Energy: 50 kcal per tablespoon (compared to mayonnaise's 594 kcal)
        * Protein: 5 g per tablespoon (compared to mayonnaise's 0.9 g)
        * Fat: 0.5 g per tablespoon (compared to mayonnaise's 64.2 g)
        * Calcium: 100 mg per tablespoon (compared to mayonnaise's 8 mg)

Greek yogurt-based spreads are another popular option, offering a tangy
flavor and high protein content. They're also lower in calories and fat
compared to traditional mayonnaise.

3. Hummus:
        * Energy: 100 kcal per tablespoon (compared to mayonnaise's 594 kcal)
        * Protein: 2 g per tablespoon (compared to mayonnaise's 0.9 g)
        * Fat: 10 g per tablespoon (compared to mayonnaise's 64.2 g)
        * Fiber: 2 g per tablespoon (compared to mayonnaise's 0 g)

Hummus is a tasty and nutritious alternative, made from chickpeas, tahini,
garlic, and lemon juice. It's rich in protein, fiber, and various vitamins
and minerals, making it a great choice for those looking for a healthier
mayonnaise substitute.

Keep in mind that these alternatives might have different flavors and
textures than traditional mayonnaise, so feel free to experiment and find the
one that works best for your taste preferences!
```

Fig. 11. Question 5 Llama 3.



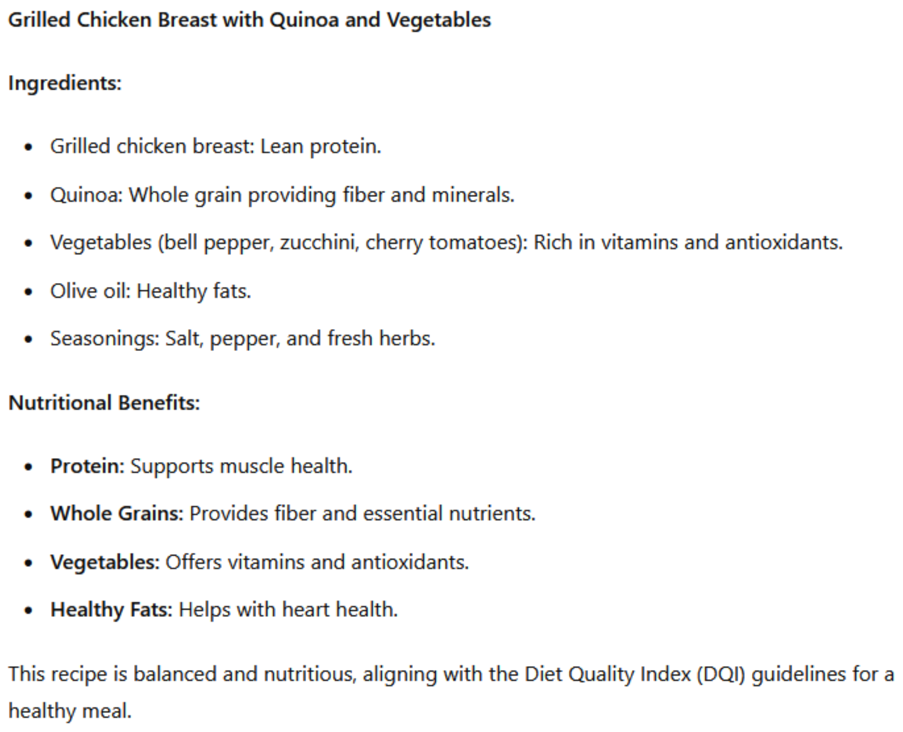
The nutritional value of a fried chicken wing can vary based on factors such as size, batter, and frying method. Here's a general estimate based on a typical serving size of one fried chicken wing (about 1.7 oz or 48 grams):

Calories: Approximately 100-110 calories
Protein: Around 7-9 grams
Total Fat: Approximately 7-9 grams

- Saturated Fat: About 2-3 grams
- Trans Fat: Varies based on frying method
  Carbohydrates: Less than 1 gram (negligible)
  Sodium: Around 130-150 mg

Keep in mind these values can vary widely depending on how the chicken wing is prepared. Factors such as whether it's breaded, how much oil is absorbed during frying, and whether any sauces are added can significantly affect its nutritional content. For a more precise value, it's best to refer to specific packaging or nutritional information from the restaurant or manufacturer.

Fig. 12. Question 1 ChatGPT3.



Grilled Chicken Breast with Quinoa and Vegetables

Ingredients:

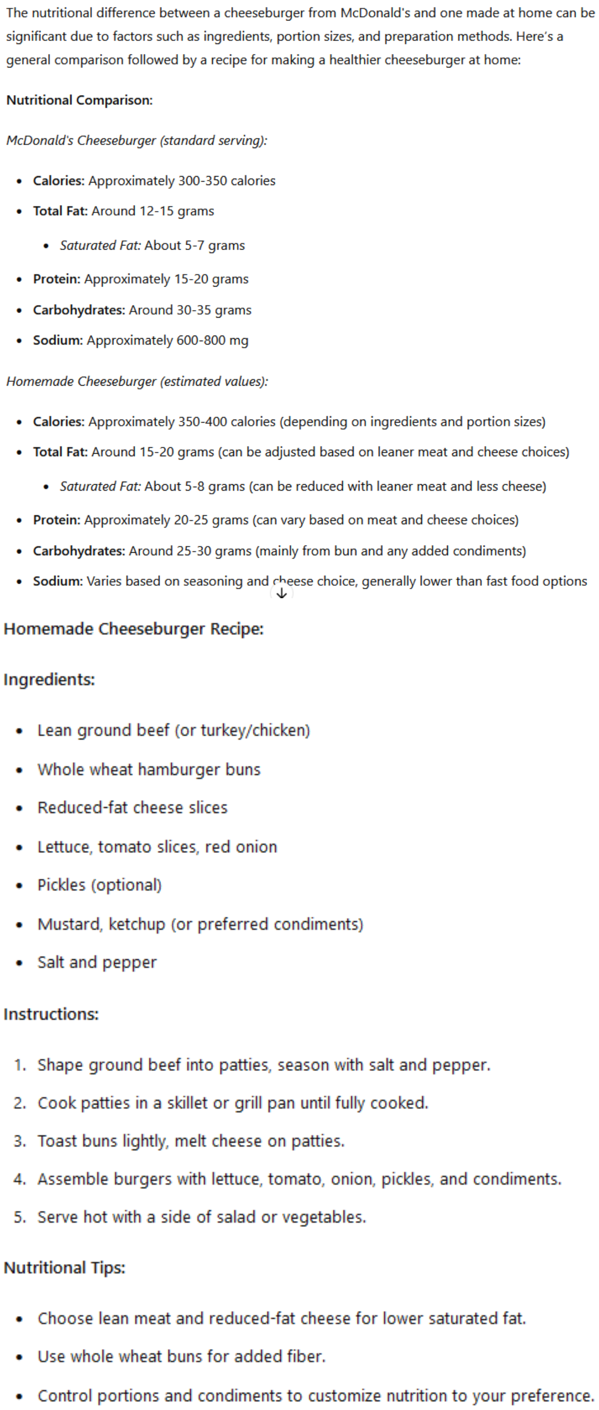
- Grilled chicken breast: Lean protein.
- Quinoa: Whole grain providing fiber and minerals.
- Vegetables (bell pepper, zucchini, cherry tomatoes): Rich in vitamins and antioxidants.
- Olive oil: Healthy fats.
- Seasonings: Salt, pepper, and fresh herbs.

Nutritional Benefits:

- **Protein:** Supports muscle health.
- **Whole Grains:** Provides fiber and essential nutrients.
- **Vegetables:** Offers vitamins and antioxidants.
- **Healthy Fats:** Helps with heart health.

This recipe is balanced and nutritious, aligning with the Diet Quality Index (DQI) guidelines for a healthy meal.

Fig. 13. Question 2 ChatGPT3.



The nutritional difference between a cheeseburger from McDonald's and one made at home can be significant due to factors such as ingredients, portion sizes, and preparation methods. Here's a general comparison followed by a recipe for making a healthier cheeseburger at home:

Nutritional Comparison:

McDonald's Cheeseburger (standard serving):

- **Calories:** Approximately 300-350 calories
- **Total Fat:** Around 12-15 grams
  - *Saturated Fat:* About 5-7 grams
- **Protein:** Approximately 15-20 grams
- **Carbohydrates:** Around 30-35 grams
- **Sodium:** Approximately 600-800 mg

Homemade Cheeseburger (estimated values):

- **Calories:** Approximately 350-400 calories (depending on ingredients and portion sizes)
- **Total Fat:** Around 15-20 grams (can be adjusted based on leaner meat and cheese choices)
  - *Saturated Fat:* About 5-8 grams (can be reduced with leaner meat and less cheese)
- **Protein:** Approximately 20-25 grams (can vary based on meat and cheese choices)
- **Carbohydrates:** Around 25-30 grams (mainly from bun and any added condiments)
- **Sodium:** Varies based on seasoning and cheese choice, generally lower than fast food options

Homemade Cheeseburger Recipe:

Ingredients:

- Lean ground beef (or turkey/chicken)
- Whole wheat hamburger buns
- Reduced-fat cheese slices
- Lettuce, tomato slices, red onion
- Pickles (optional)
- Mustard, ketchup (or preferred condiments)
- Salt and pepper

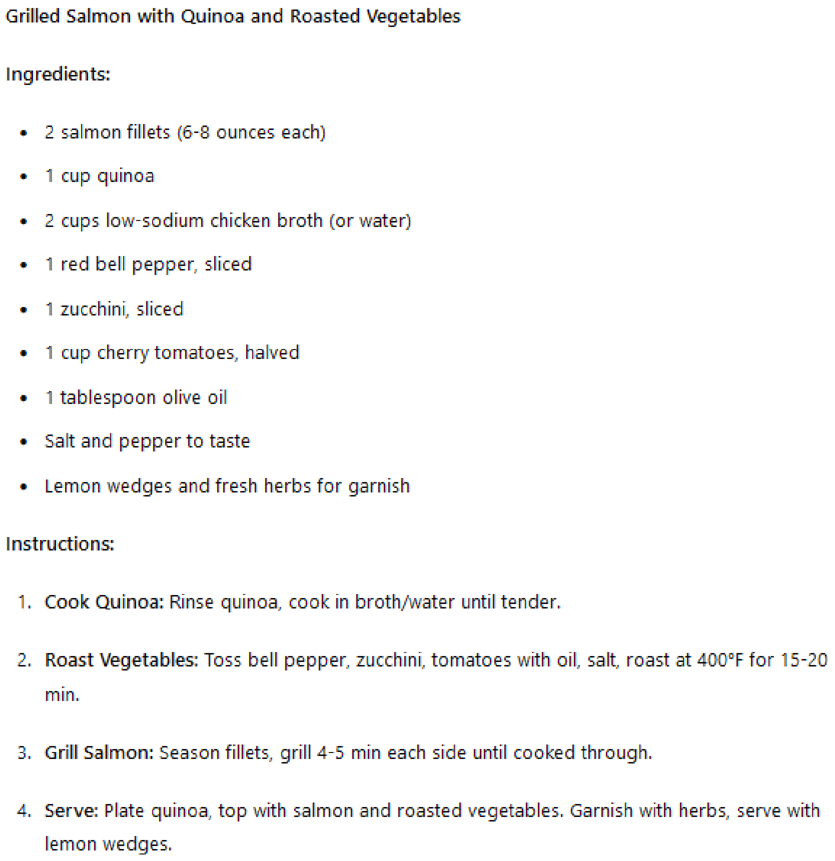
Instructions:

1. Shape ground beef into patties, season with salt and pepper.
2. Cook patties in a skillet or grill pan until fully cooked.
3. Toast buns lightly, melt cheese on patties.
4. Assemble burgers with lettuce, tomato, onion, pickles, and condiments.
5. Serve hot with a side of salad or vegetables.

Nutritional Tips:

- Choose lean meat and reduced-fat cheese for lower saturated fat.
- Use whole wheat buns for added fiber.
- Control portions and condiments to customize nutrition to your preference.

Fig. 14. Question 3 ChatGPT3.



Grilled Salmon with Quinoa and Roasted Vegetables

Ingredients:

- 2 salmon fillets (6-8 ounces each)
- 1 cup quinoa
- 2 cups low-sodium chicken broth (or water)
- 1 red bell pepper, sliced
- 1 zucchini, sliced
- 1 cup cherry tomatoes, halved
- 1 tablespoon olive oil
- Salt and pepper to taste
- Lemon wedges and fresh herbs for garnish

Instructions:

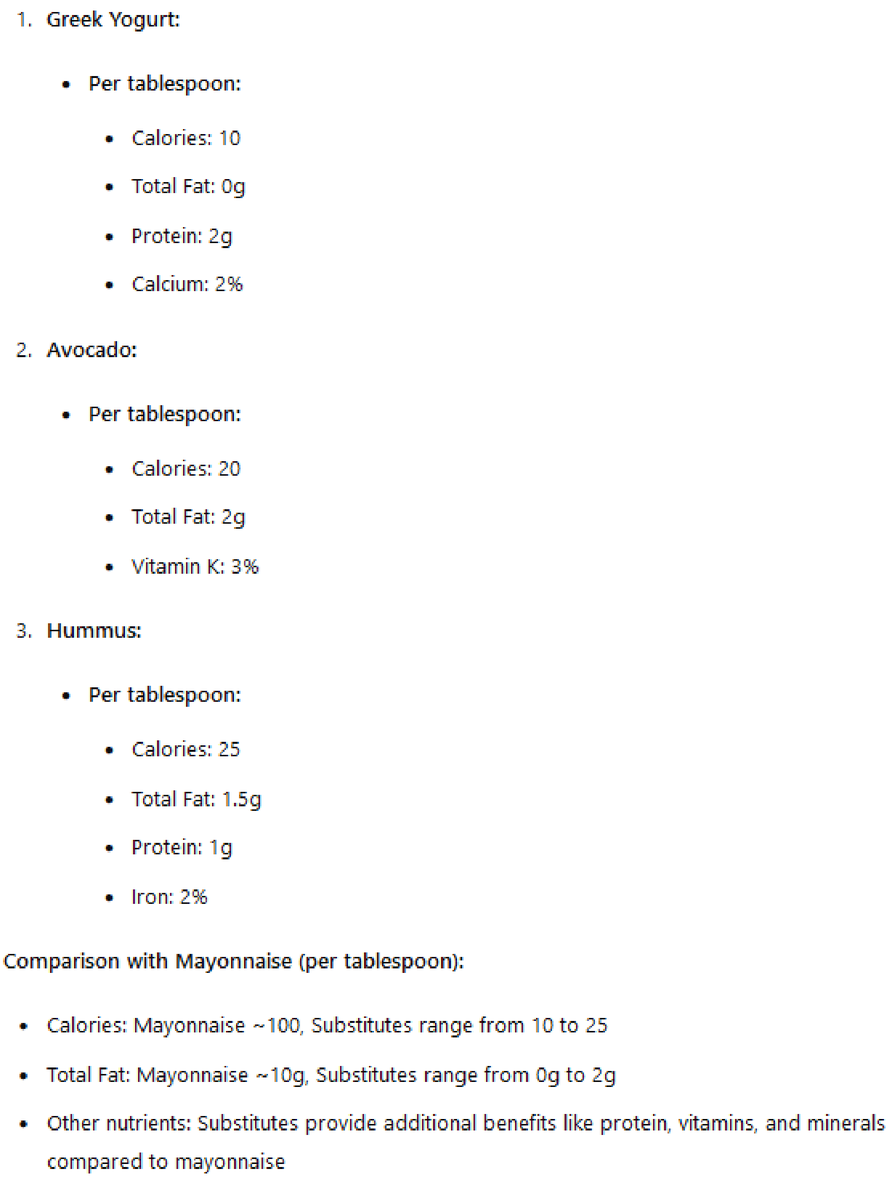
1. **Cook Quinoa:** Rinse quinoa, cook in broth/water until tender.
2. **Roast Vegetables:** Toss bell pepper, zucchini, tomatoes with oil, salt, roast at 400°F for 15-20 min.
3. **Grill Salmon:** Season fillets, grill 4-5 min each side until cooked through.
4. **Serve:** Plate quinoa, top with salmon and roasted vegetables. Garnish with herbs, serve with lemon wedges.

Fig. 15. Question 4 ChatGPT3.

This highlighted the differences between models with Retrieval-Augmented Generation (RAG) and those without. Utilizing retrieved information, Llama-3 minimized the common issue of hallucinations in large language models (LLMs), resulting in more accurate responses. Moreover, the model discerned whether queried ingredients or terms were relevant to the context of recipe questions, ensuring contextual relevance in its answers.

In question 1 of Llama 3, we observed its capability to retrieve information specifically about a pre-cooked fried chicken wing identified by the food code 24167210. While the model typically does not mention the food code in its responses, the context printed indicates successful retrieval of information from the data used in Retrieval-Augmented Generation (RAG). Below is an example of the context provided for the question:

- Food code: 24167210
- Main food description: Chicken wing, fried, coated, from pre-cooked
- WWEIA Category number: 2202



1. **Greek Yogurt:**
   - **Per tablespoon:**
     - Calories: 10
     - Total Fat: 0g
     - Protein: 2g
     - Calcium: 2%
2. **Avocado:**
   - **Per tablespoon:**
     - Calories: 20
     - Total Fat: 2g
     - Vitamin K: 3%
3. **Hummus:**
   - **Per tablespoon:**
     - Calories: 25
     - Total Fat: 1.5g
     - Protein: 1g
     - Iron: 2%

Comparison with Mayonnaise (per tablespoon):

- Calories: Mayonnaise ~100, Substitutes range from 10 to 25
- Total Fat: Mayonnaise ~10g, Substitutes range from 0g to 2g
- Other nutrients: Substitutes provide additional benefits like protein, vitamins, and minerals compared to mayonnaise

Fig. 16. Question 5 ChatGPT3.

- WWEIA Category description: Chicken, whole pieces
- Energy (kcal): 294
- Protein (g): 18.62
- Carbohydrate (g): 7.15

- Sugars, total (g): 0.12
- Fiber, total dietary (g): 0.3
- Total Fat (g): 20.59
- Fatty acids, total saturated (g): 4.793

This example demonstrates the successful functioning of the vector retriever, which retrieves different foods for each question's context. In contrast, Llama 2 retrieved various types of fried chicken wings, differing mainly in cooking methods or spices, thereby providing a range of nutritional values instead of the specific values as detailed by Llama 3.

5 Conclusion

Large Language Models (LLMs) can serve various purposes by being fine-tuned for specific tasks and offering a user-friendly interface, making them accessible to individuals across different age groups. By training these models with food-related data, we observed their capability to retrieve and utilize information effectively. Such LLMs can significantly assist in addressing the challenge of meal planning on a daily basis, while also facilitating the attainment of necessary daily macronutrients and micronutrients. Future efforts will focus on refining our model through various fine-tuning approaches to enhance accuracy and precision. Additionally, we plan to develop databases tailored to specific dietary needs and recipes from diverse cultural backgrounds, and to expand our nutritional data beyond the USA. Comparative studies involving other models will be conducted to identify optimal performance, and we aim to integrate voice interaction capabilities into these models to enhance accessibility for users unfamiliar with technology.

References

1. Aljbawi, B.: Health-aware food planner: A personalized recipe generation approach based on GPT-2. Ph.D. thesis, theses and dissertations (2020) scholars.wlu.ca/etd/2311
2. Campos, S., Doxey, J., Hammond, D.: Nutrition labels on pre-packaged foods: A systematic review. Public Health Nutrition, vol. 14, no. 8, pp. 1496–1506 (2011) doi: 10.1017/S1368980010003290
3. Deng, Y., Zhao, N., Huang, X.: Early ChatGPT user portrait through the lens of data. In: Proceedings of the IEEE International Conference on Big Data (BigData), pp. 4770–4775 (2023) doi: 10.1109/bigdata59044.2023.10386415
4. Grunert, K. G., Wills, J. M., Fernández-Celemín, L.: Nutrition knowledge, and use and understanding of nutrition information on food labels among consumers in the UK. Appetite, vol. 55, no. 2, pp. 177–189 (2010) doi: 10.1016/j.appet.2010.05.045
5. Kim, S., Haines, P. S., Siega-Riz, A. M., Popkin, B. M.: The diet quality index-international (DQI-I) provides an effective tool for cross-national comparison of diet quality as illustrated by China and the United States. The Journal of Nutrition, vol. 133, no. 11, pp. 3476–3484 (2003) doi: 10.1093/jn/133.11.3476

6. Meta AI: Introducing Meta Llama 3: The most capable openly available LLM to date (2024) ai.meta.com/blog/meta-llama-3/
7. Naeeni, M., Jafari, S., Fouladgar, M., Heidari, K., Farajzadegan, Z., Fakhri, M., Karami, P., Omidi, R.: Nutritional knowledge, practice, and dietary habits among school children and adolescents. International Journal of Preventive Medicine, vol. 5, no. 14, pp. 171 (2014) doi: 10.4103/2008-7802.157687
8. OpenAI: GPT-4 technical report. Technical report, OpenAI (2023)
9. Ricciardelli, L. A., McCabe, M. P.: Children's body image concerns and eating disturbance. Clinical Psychology Review, vol. 21, no. 3, pp. 325–344 (2001) doi: 10.1016/s0272-7358(99)00051-3
10. Saeedi, P., Petersohn, I., Salpea, P., Malanda, B., Karuranga, S., Unwin, N., Colagiuri, S., Guariguata, L., Motala, A. A., Ogurtsova, K., Shaw, J. E., Bright, D., Williams, R.: Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the international diabetes federation diabetes atlas, 9th edition. Diabetes Research and Clinical Practice, vol. 157, pp. 107843 (2019) doi: 10.1016/j.diabres.2019.107843
11. Shepherd, J., Harden, A., Rees, R., Brunton, G., Garcia, J., Oliver, S., Oakley, A.: Young people and healthy eating: A systematic review of research on barriers and facilitators. Health Education Research, vol. 21, no. 2, pp. 239–257 (2005) doi: 10.1093/her/cyh060
12. Teng, C. Y., Lin, Y. R., Adamic, L. A.: Recipe recommendation using ingredient networks. In: Proceedings of the 4th Annual ACM Web Science Conference, pp. 298–307 (2012) doi: 10.1145/2380718.2380757
13. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., et al.: Llama 2: Open foundation and fine-tuned chat models (2023) doi: 10.48550/arXiv.2307.09288
14. Zhang, J., Li, M., Liu, W., Lauria, S., Liu, X.: Many-objective optimization meets recommendation systems: A food recommendation scenario. Neurocomputing, vol. 503, pp. 109–117 (2022) doi: 10.1016/j.neucom.2022.06.081

ISSN 1870-4069

LLAMA Assisted Nutritional Recipe Suggestions: Integrating the Dietary Quality Index for Health Conscious Cooking

Diego Estrada-Beltrán, Miguel Gonzalez-Mendoza,
Raul Monroy-Borja, Gilberto Ochoa-Ruiz,
Janet Gutiérrez-Uribe, Astrid Domínguez-Uscanga

Tecnológico de Monterrey,
Escuela de Ingeniería y Ciencias,
Mexico

{a01252679, mgonza, raulm, gilberto.ochoa, jagu,
astrid.dominguez}@tec.mx

Abstract. Choosing what to eat daily can be challenging due to the variety of ingredients and their differing nutritional values, often leading to health problems such as cardiovascular diseases and psychological disorders. The recent surge in popularity of Large Language Models (LLMs) like ChatGPT, Gemini, and Llama has made it easier for users to obtain information and recommendations from various fields more quickly than traditional online searches. In this study, we trained Llama models using datasets on ingredients and their nutritional values, fast food menus with nutritional values, and the Dietary Quality Index (DQI) to provide personalized recipe suggestions with accurate nutritional information. We compare the performance of Retrieval Augmented Generation (RAG) with Llama-2 model, Llama-3, and the default ChatGPT model in terms of recipe accuracy, nutritional value precision, and user-friendliness. Our findings aim to demonstrate the potential of LLMs in improving dietary health through accessible and understandable nutritional guidance, while also addressing the common problem of hallucination in LLMs by using RAG for context information.

Keywords: DQI, Llama, LLM, RAG, nutritional daily intake, recipe recommendation.

1 Introduction

In kitchens around the globe, a daily dilemma unfolds: “What should I cook today?” With the abundance of recipes and ingredients making a single meal becomes a difficult task, and, beyond this problem, nutritional values of the food are another problem to take in account. With the lack of nutritional knowledge and the lack of knowledge to interpret the information provided by nutrition labels for most people [2, 4], this dilemma becomes hard to answer. In 1990, the Nutrition Labeling and Education Act (NLEA) was introduced in the United States, marking a pivotal moment in public health policy.

Its primary objective was to give consumers the necessary tools to make informed dietary choices. By using standardized labels with essential information, people could opt for healthier products. However, despite the widespread adoption of nutritional labels, research indicates that a substantial portion of the population overlooks or struggles to interpret these labels, Campos et al. (2011) showed that nutrition labels that require calculations concerning nutrient amounts and serving sizes are confusing to many consumers, particularly those with lower education and literacy skills so, a graphical view would be more helpful. Grunert et al. (2010) highlighted that from a questionnaire of 921 people, 27% checked the nutritional labels and from this, 70 to 90% could correctly interpret the information.

This lack of nutritional understanding can lead to imbalanced dietary patterns [7], characterized by both excessive and deficient consumption of certain nutrients. Such dietary imbalances are associated with a spectrum of non-communicable diseases (NCDs), encompassing immediate complications and long-term health consequences, including cardiovascular diseases, diabetes, hypertension, stroke, cancer, dental caries, asthma, and various psychological disorders such as depression [9, 11].

Alarmingly, the International Diabetes Federation reports that approximately 415 million people worldwide suffer from diabetes, with incidence rates projected to surge by over 50% by 2040. Furthermore, the Global Burden of Disease Study underscores the significant contribution of dietary factors to levels of malnutrition, obesity, and overweight, with 11 million preventable premature deaths annually attributed to unreasonable dietary habits [10].

The advancements in machine learning have spurred the development of numerous recommendation systems (RSs) aimed at addressing challenges in recipe suggestions and nutritional knowledge. For instance, Zhang et al (2022) introduced the MaOO model, a vector optimization algorithm designed to optimize multiple objective functions [14]. By seamlessly integrating ingredients' nutritional value, food diversity, and user dietary patterns, the MaOO model correctly suggests suitable food options for users. Similarly, Teng et al. (2012) proposed a methodology leveraging pointwise mutual information (PMI) coupled with support vector machines (SVMs).

This approach establishes correlations among ingredients found in diverse recipes, identifying the most frequently used ingredients. By employing SVMs, it helped to further refine recipe suggestions [12]. or the most akin to this work, introduced by Aljbawi Bushra (2020) who utilized GPT-2 to generate recipe suggestions based on input prompts containing ingredients by training it with different recipes and the ingredients used [1].

The introduction of OpenAI's GPT-3 public model in November 2022 has sparked a surge in interest surrounding Generative Artificial Intelligence (GAI) models. With reports indicating that over 100 million individuals engaged with the model in January 2023 alone [3], GAI technologies have become a focal point of contemporary discourse. Utilizing Generative Pre-trained Transformers (GPT), these models are trained on extensive textual datasets, empowering them to produce coherent and contextually relevant responses. Notably, the GPT-3 model boasts an impressive 175 billion parameters, derived from textual data converted into tokens.

Subsequently, based on user prompts, a retriever mechanism identifies and retrieves the most pertinent information to address user inquiries [8]. As a result of the widespread acclaim garnered by the GPT model, numerous other companies have ventured into the development of their own Generative Artificial Intelligence (GAI) models. Among these is Meta's Llama-2 model, introduced in July 2023 [13]. A refined iteration of the original Llama 1 model, Llama-2 has been trained on extensive datasets, resulting in versions with parameters ranging from 7 billion to 70 billion.

On April 18, 2024, Meta released a new model named Llama-3, with parameters ranging from 8 to 70 billion. It was trained on 15 trillion tokens from public sources, making its training dataset seven times larger than that of Llama-2. Additionally, Llama-3 utilizes a tokenizer with 128,000 tokens to encode language more efficiently. Meta also modified the code to enhance the model's manipulability [6].

Capitalizing on the growing popularity of these models and recognizing the imperative to address malnutrition risks, we have developed an innovative model with enhanced user interaction. This model not only provides recipe suggestions based on available ingredients but also offers personalized nutritional guidance tailored to individual needs. Equipped with memory functionality, our model dynamically adapts to users' dietary preferences, leveraging Dietary Quality Index (DQI) scores to monitor and adjust nutritional intake. This adaptive approach ensures the delivery of healthy recipe suggestions suitable for individuals of all ages and varying culinary skills.

Building upon these capabilities, our model goes a step further by offering recipe suggestions based on portion sizes, enabling more precise monitoring of nutritional intake. This enhanced feature ensures a more controlled approach to recipe recommendations, facilitating accurate tracking of nutrition and promoting healthier dietary habits.

2 Materials and Methods

2.1 Data Collection

We collected data through web scraping from two primary sources: FoodData Central USDA and MenuWithNutrition. From FoodData Central USDA, we obtained a dataset comprising ingredients along with their corresponding nutritional values. The MenuWithNutrition website provided us with a diverse dataset of USA fast food menus, including detailed nutritional information for each menu item. Additionally, to ensure that our model recommends recipes aligned with nutritional guidelines, we integrated data from the Dietary Quality Index (DQI) document.

This document informed our model to prioritize recipes that contribute to achieving the nutritional scores outlined in the DQI. For the data extraction process, we utilized the request library to access the respective websites. Specifically, urllib.request and urlopen were employed to fetch data from the URLs of these websites. Subsequently, BeautifulSoup was utilized to parse the HTML structure and extract the relevant information needed from the web pages.

Table 1. Fraction of USDA dataset.

Main food description	Food code	WWEIA Number	WWEIA Description	Energy (kcal)	Protein (g)
Milk, human	11000000	9602	Human milk	70	1.03
Milk, NFS	11100000	1004	Milk, reduced fat	52	3.33
Milk, whole	11111000	1002	Milk, whole	61	3.27
Milk, reduced fat (2%)	11112110	1004	Milk, reduced fat	50	3.36
Milk, low fat (1%)	11112210	1006	Milk, lowfat	43	3.38

2.2 Data Understanding

FoodData Central USDA. FoodData Central (FDC) is a comprehensive food and nutrient database maintained by the United States Department of Agriculture (USDA), offering detailed information on the nutritional composition of a wide array of foods consumed in the United States. This database encompasses essential data on nutrients such as vitamins, minerals, carbohydrates, proteins, fats, and other components present in foods. From FDC, we utilized the Food and Nutrient Database for Dietary Studies 2019-2020 (FNDDS 2019-2020). This specialized database provides nutritional details for foods and beverages reported in ”What We Eat in America,” a component of the National Health and Nutrition Examination Survey (NHANES). This table, with 5625 rows and 69 columns, contains food descriptions and their respective food codes from the FNDDS.

Each 8-digit food code starts with a digit representing one of nine major food groups: Milk and Milk Products, Meat, Poultry, Fish, and Mixtures, Eggs, Dry Beans, Peas, Other Legumes, Nuts, and Seeds, Grain Products, Fruits, Vegetables, Fats, Oils, and Salad Dressings, Sugars, Sweets, and Beverages. The remaining digits specify subgroups, with codes ranging from 11000000 to 99998210. It includes WWEIA Category numbers and descriptions for 170 subgroups, ranging from 1002 to 9999, which classify foods into specific subgroups. The rest of the columns provide nutritional values (micro and macronutrients) per 100 grams of each food item.

MenuWithNutrition. MenuWithNutrition is a website dedicated to providing detailed information on menus from various fast food chains across the USA. They compile data from multiple sources and meticulously verify the accuracy of nutritional values. The website is designed to be comprehensive and user-friendly, helping individuals understand the nutritional content of food items from different restaurant chains. From MenuWithNutrition, we obtained a database similar to that of the USDA. This database includes information on restaurants, menu items, and nutritional values such as macronutrients (fats, proteins, carbohydrates, fiber) and micronutrients (monounsaturated fats, polyunsaturated fats, among others).

Dietary Quality Index. The Dietary Quality Index (DQI) is a metric developed by Soowon Kim et al. [5] to evaluate the overall quality of daily food intake. It is structured into four key categories aimed at assessing different nutritional aspects:

1. **Variety:** This category evaluates the diversity of food sources within the diet, including proteins from various food groups such as meats, fruits, vegetables, dairy, and grains.

Table 2. Fraction of MenuWithNutrition dataset.

Restaurant Name	Food Name	Total Fat	Cholesterol	Sodium	Total Carbohydrate	Protein
aandw restaurant	chocolate cone	4.5 g	15 mg	105 g	26 g	3 g
aandw restaurant	root beer float	5.2 g	39 mg	104 g	70.4 g	2.1 g
aandw restaurant	diet root beer float	5.2 g	39 mg	104 g	31.1 g	2.1 g
aandw restaurant	root beer freeze	18 g	70 mg	400 g	150 g	16 g
aandw restaurant	chocolate shake	28.8 g	124 mg	200 g	100 g	11.2 g

2. **Adequacy:** Adequacy assesses whether the intake of essential dietary elements meets recommended levels to ensure a healthy diet and prevent undernutrition.
3. **Moderation:** This examines the intake of foods and nutrients associated with chronic diseases, emphasizing moderation of total fats, saturated fats, cholesterol, and sodium.
4. **Overall Balance:** The final category evaluates the overall balance of the diet in terms of energy sources and fatty acid composition, aiming for proportional intake across these categories.

2.3 Llama and Retrieval Augmented Generation

The Llama-2 and Llama-3 models, with 7 billion and 8 billion parameters respectively, were used as the foundation for our approach. These models, developed by Meta, were downloaded from Hugging Face. To fine-tune and manipulate these models, we utilized the Langchain library, known for its versatile capabilities in model adaptation. An essential component of our method was the system prompt template, which guided the model's responses. The template used was as follows:

- “”Use the following pieces of information to answer the user's question. If you don't know the answer, just say that you don't know, don't try to make up an answer.
- Context: context.
- Question: question.

You task is to suggest recipes and give people information about nutritional information about their food based on the Dietary Quality Index (DQI). The necessary nutritional values to take into account for the DQI and the recipe suggestions are: Protein (g), Lipids (g), Fiber (g), Ascorbic Acid (mg), Cholesterol (mg), Saturated Fatty Acids (g), Calcium (mg), Iron (mg), Sodium (mg), Carbohydrates (g), SFAs, MFAs, PUFAs and Total Energy (Kcal).

- Helpful Answer: ””

This template ensures the model answers questions only if it knows the answer. Also, since it is a safe mode, it doesn't respond to harmful queries, thus eliminating the need for additional safety instructions in the template.

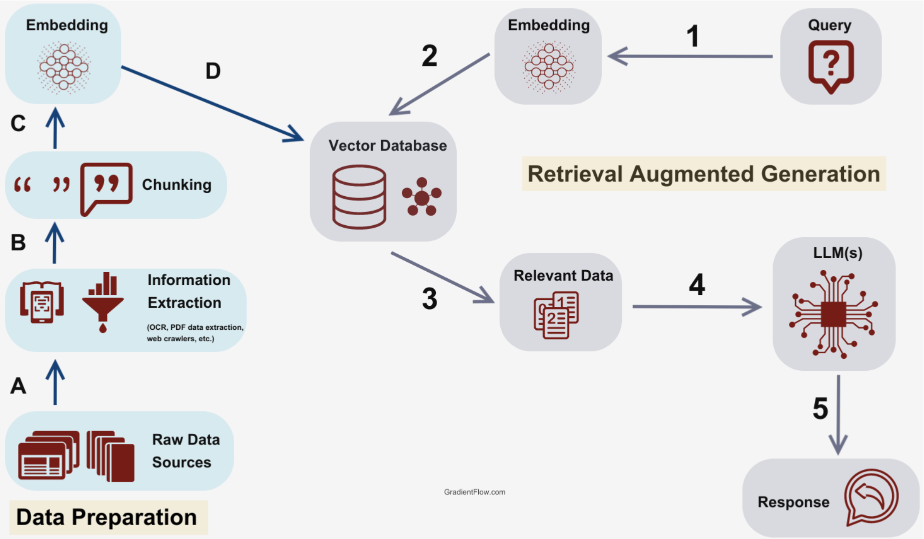


Fig. 1. Retrieval-augmented generation.

By including the context and question parts, we enable the model to retrieve relevant information from the uploaded data to answer questions accurately. The task specified in the template directs the model to suggest recipes and provide their nutritional information, aligning the model's responses with our objectives. To prepare our data for the model, we performed ETL (Extract, Transform, Load) operations. Initially, we extracted comprehensive data through web scraping. Next, we cleaned the data by removing null and duplicate values to ensure data integrity and consistency. Finally, we loaded the cleaned data into the model.

The cleaned datasets were then formatted into .csv files, facilitating seamless integration with the Langchain RAG technique. The RAG (Retrieval-Augmented Generation) technique served as a pivotal component in augmenting the Llama models' knowledge with additional data. This technique comprises five key steps: Load, Split, Store, Retrieve, and Generate. Through this process, we were able to effectively incorporate diverse datasets into the model, enriching its understanding of nutritional information and enhancing its capacity to generate relevant recipe suggestions based on user queries:

1. **Load:** The initial step of the RAG technique involves loading the three datasets we possess. Langchain, our chosen tool for this task, is capable of accepting various document formats such as PDFs, CSV files, SQL databases, or even website text in HTML format.

2. **Split:** Given that LLM models have constraints regarding the length of input they can process, we implement a splitting process for our documents. Each document is divided into manageable chunks, with each chunk containing approximately 1000 characters. To maintain continuity and coherence, these chunks overlap by 200

characters from one document to the next. This approach ensures that the model can follow the sequential order of the documents while processing them.

3. **Store:** Following the splitting process, we transform these chunks into embeddings, which are numerical representations of the text data. These embeddings are then stored in a vector database, which we opted to utilize the Chroma vector database for this purpose. Storing the embeddings in a vector database facilitates efficient retrieval based on similarity search, a crucial aspect of the subsequent steps.
4. **Retrieve:** Using the stored embeddings in the vector database, we perform similarity searches to retrieve the most relevant embeddings corresponding to the prompt provided by the user. This retrieval process ensures that the model accesses pertinent information from the datasets that aligns closely with the user's query, enhancing the relevance and accuracy of the model's responses.
5. **Generate:** The final step of the RAG technique involves the generation of responses by the model. This process unfolds as a chain, beginning with the user's prompt, followed by the retrieval of relevant information from the stored embeddings. The model then constructs a response based on this retrieved information, synthesizing it with the user's prompt to generate a coherent and contextually relevant output.

3 Results

To evaluate the models' performance, we used the following questions to assess their ability to generate creative and accurate responses, verify the accuracy of nutritional values, and correctly retrieve data for contextual understanding:

1. Retrieve nutritional information for a food item (e.g., chicken wing).
2. Provide a recipe that meets the necessary nutritional intake according to the Dietary Quality Index (DQI).
3. Explain the differences between fast food and homemade food.
4. Create a recipe centered around a specific ingredient.
5. Compare foods to identify healthier meal options, such as suggesting substitutes for mayonnaise in sandwiches.

3.1 Llama-2 7b Answers

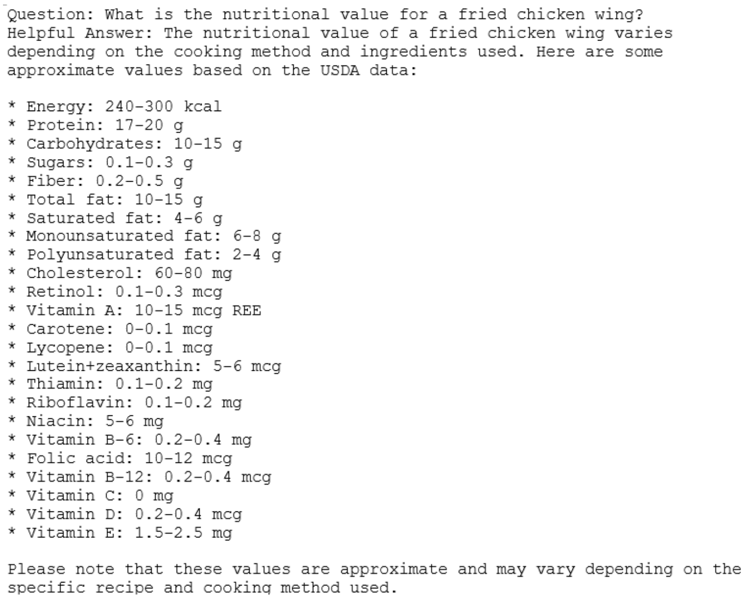
Shown from Figure 2 to Figure 6

3.2 Llama-3 8b Answers

Shown from Figure 7 to Figure 11

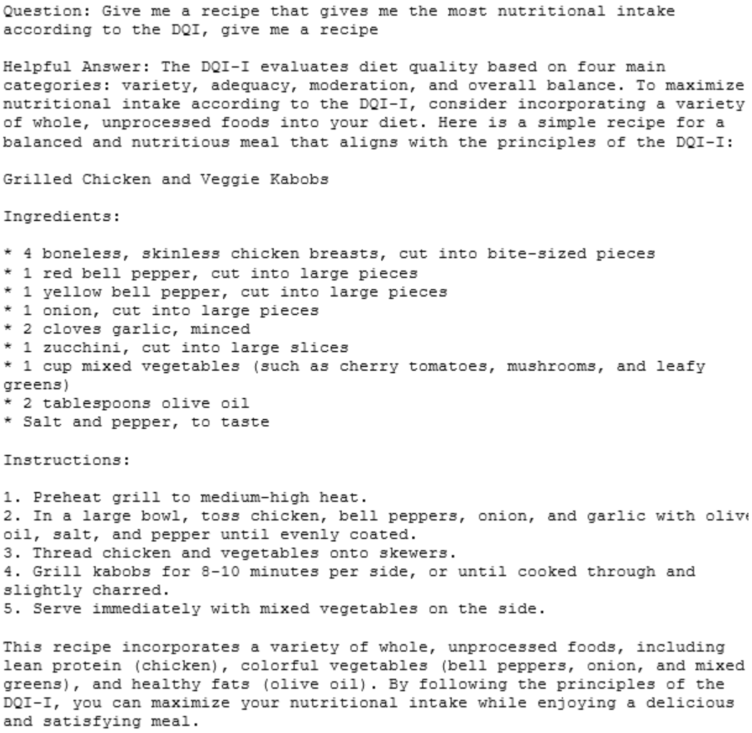
3.3 Default ChatGPT-3 Answers

Shown from Figure 12 to Figure 16



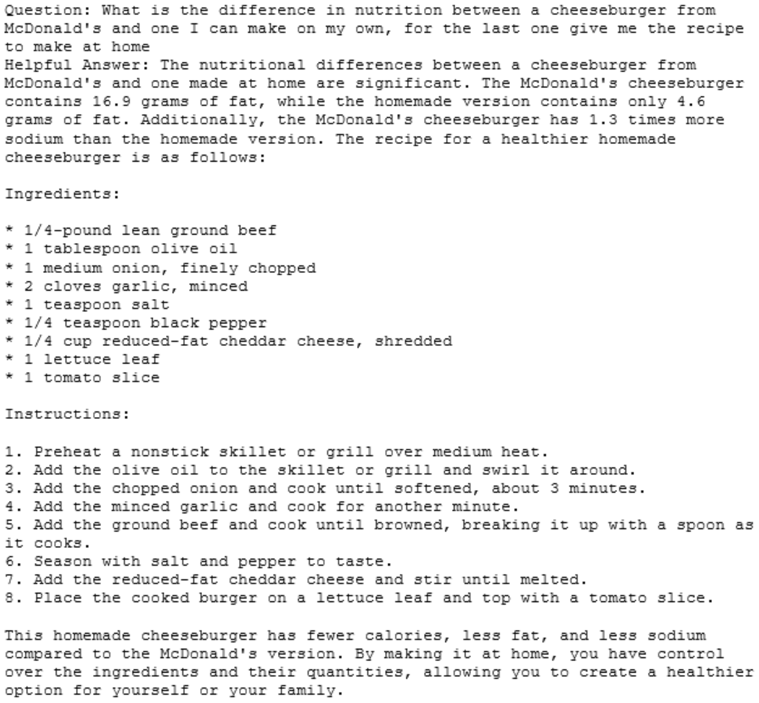
```
Question: What is the nutritional value for a fried chicken wing?
Helpful Answer: The nutritional value of a fried chicken wing varies
depending on the cooking method and ingredients used. Here are some
approximate values based on the USDA data:

* Energy: 240-300 kcal
* Protein: 17-20 g
* Carbohydrates: 10-15 g
* Sugars: 0.1-0.3 g
* Fiber: 0.2-0.5 g
* Total fat: 10-15 g
* Saturated fat: 4-6 g
* Monounsaturated fat: 6-8 g
* Polyunsaturated fat: 2-4 g
* Cholesterol: 60-80 mg
* Retinol: 0.1-0.3 mcg
* Vitamin A: 10-15 mcg REE
* Carotene: 0-0.1 mcg
* Lycopene: 0-0.1 mcg
* Lutein+zeaxanthin: 5-6 mcg
* Thiamin: 0.1-0.2 mg
* Riboflavin: 0.1-0.2 mg
* Niacin: 5-6 mg
* Vitamin B-6: 0.2-0.4 mg
* Folic acid: 10-12 mcg
* Vitamin B-12: 0.2-0.4 mcg
* Vitamin C: 0 mg
* Vitamin D: 0.2-0.4 mcg
* Vitamin E: 1.5-2.5 mg

Please note that these values are approximate and may vary depending on the
specific recipe and cooking method used.
```

Fig. 2. Question 1 Llama 2.



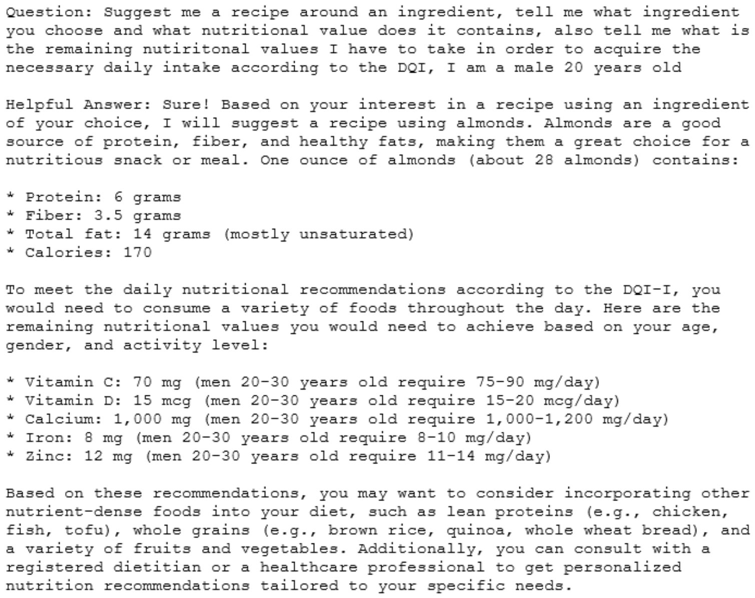
```
Question: Give me a recipe that gives me the most nutritional intake
according to the DQI, give me a recipe

Helpful Answer: The DQI-I evaluates diet quality based on four main
categories: variety, adequacy, moderation, and overall balance. To maximize
nutritional intake according to the DQI-I, consider incorporating a variety
of whole, unprocessed foods into your diet. Here is a simple recipe for a
balanced and nutritious meal that aligns with the principles of the DQI-I:

Grilled Chicken and Veggie Kabobs

Ingredients:

* 4 boneless, skinless chicken breasts, cut into bite-sized pieces
* 1 red bell pepper, cut into large pieces
* 1 yellow bell pepper, cut into large pieces
* 1 onion, cut into large pieces
* 2 cloves garlic, minced
* 1 zucchini, cut into large slices
* 1 cup mixed vegetables (such as cherry tomatoes, mushrooms, and leafy
greens)
* 2 tablespoons olive oil
* Salt and pepper, to taste

Instructions:

1. Preheat grill to medium-high heat.
2. In a large bowl, toss chicken, bell peppers, onion, and garlic with oliv
oil, salt, and pepper until evenly coated.
3. Thread chicken and vegetables onto skewers.
4. Grill kabobs for 8-10 minutes per side, or until cooked through and
slightly charred.
5. Serve immediately with mixed vegetables on the side.

This recipe incorporates a variety of whole, unprocessed foods, including
lean protein (chicken), colorful vegetables (bell peppers, onion, and mixed
greens), and healthy fats (olive oil). By following the principles of the
DQI-I, you can maximize your nutritional intake while enjoying a delicious
and satisfying meal.
```

Fig. 3. Question 2 Llama 2.



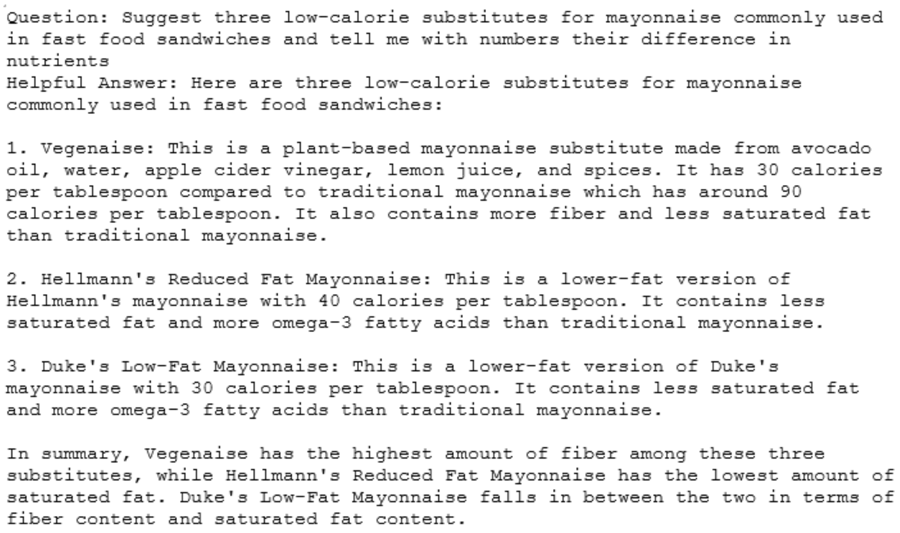
```
Question: What is the difference in nutrition between a cheeseburger from
McDonald's and one I can make on my own, for the last one give me the recipe
to make at home
Helpful Answer: The nutritional differences between a cheeseburger from
McDonald's and one made at home are significant. The McDonald's cheeseburger
contains 16.9 grams of fat, while the homemade version contains only 4.6
grams of fat. Additionally, the McDonald's cheeseburger has 1.3 times more
sodium than the homemade version. The recipe for a healthier homemade
cheeseburger is as follows:

Ingredients:

* 1/4-pound lean ground beef
* 1 tablespoon olive oil
* 1 medium onion, finely chopped
* 2 cloves garlic, minced
* 1 teaspoon salt
* 1/4 teaspoon black pepper
* 1/4 cup reduced-fat cheddar cheese, shredded
* 1 lettuce leaf
* 1 tomato slice

Instructions:

1. Preheat a nonstick skillet or grill over medium heat.
2. Add the olive oil to the skillet or grill and swirl it around.
3. Add the chopped onion and cook until softened, about 3 minutes.
4. Add the minced garlic and cook for another minute.
5. Add the ground beef and cook until browned, breaking it up with a spoon as
it cooks.
6. Season with salt and pepper to taste.
7. Add the reduced-fat cheddar cheese and stir until melted.
8. Place the cooked burger on a lettuce leaf and top with a tomato slice.

This homemade cheeseburger has fewer calories, less fat, and less sodium
compared to the McDonald's version. By making it at home, you have control
over the ingredients and their quantities, allowing you to create a healthier
option for yourself or your family.
```

Fig. 4. Question 3 Llama 2.



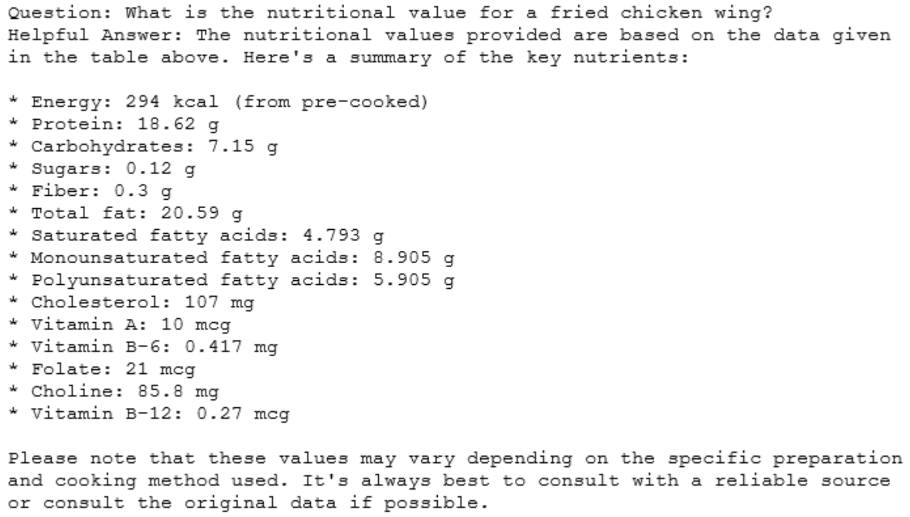
```
Question: Suggest me a recipe around an ingredient, tell me what ingredient
you choose and what nutritional value does it contains, also tell me what is
the remaining nutiritonal values I have to take in order to acquire the
necessary daily intake according to the DQI, I am a male 20 years old

Helpful Answer: Sure! Based on your interest in a recipe using an ingredient
of your choice, I will suggest a recipe using almonds. Almonds are a good
source of protein, fiber, and healthy fats, making them a great choice for a
nutritious snack or meal. One ounce of almonds (about 28 almonds) contains:

* Protein: 6 grams
* Fiber: 3.5 grams
* Total fat: 14 grams (mostly unsaturated)
* Calories: 170

To meet the daily nutritional recommendations according to the DQI-I, you
would need to consume a variety of foods throughout the day. Here are the
remaining nutritional values you would need to achieve based on your age,
gender, and activity level:

* Vitamin C: 70 mg (men 20-30 years old require 75-90 mg/day)
* Vitamin D: 15 mcg (men 20-30 years old require 15-20 mcg/day)
* Calcium: 1,000 mg (men 20-30 years old require 1,000-1,200 mg/day)
* Iron: 8 mg (men 20-30 years old require 8-10 mg/day)
* Zinc: 12 mg (men 20-30 years old require 11-14 mg/day)

Based on these recommendations, you may want to consider incorporating other
nutrient-dense foods into your diet, such as lean proteins (e.g., chicken,
fish, tofu), whole grains (e.g., brown rice, quinoa, whole wheat bread), and
a variety of fruits and vegetables. Additionally, you can consult with a
registered dietitian or a healthcare professional to get personalized
nutrition recommendations tailored to your specific needs.
```

Fig. 5. Question 4 Llama 2.



```
Question: Suggest three low-calorie substitutes for mayonnaise commonly used
in fast food sandwiches and tell me with numbers their difference in
nutrients
Helpful Answer: Here are three low-calorie substitutes for mayonnaise
commonly used in fast food sandwiches:

1. Vegenaise: This is a plant-based mayonnaise substitute made from avocado
oil, water, apple cider vinegar, lemon juice, and spices. It has 30 calories
per tablespoon compared to traditional mayonnaise which has around 90
calories per tablespoon. It also contains more fiber and less saturated fat
than traditional mayonnaise.

2. Hellmann's Reduced Fat Mayonnaise: This is a lower-fat version of
Hellmann's mayonnaise with 40 calories per tablespoon. It contains less
saturated fat and more omega-3 fatty acids than traditional mayonnaise.

3. Duke's Low-Fat Mayonnaise: This is a lower-fat version of Duke's
mayonnaise with 30 calories per tablespoon. It contains less saturated fat
and more omega-3 fatty acids than traditional mayonnaise.

In summary, Vegenaise has the highest amount of fiber among these three
substitutes, while Hellmann's Reduced Fat Mayonnaise has the lowest amount of
saturated fat. Duke's Low-Fat Mayonnaise falls in between the two in terms of
fiber content and saturated fat content.
```

Fig. 6. Question 5 Llama 2.

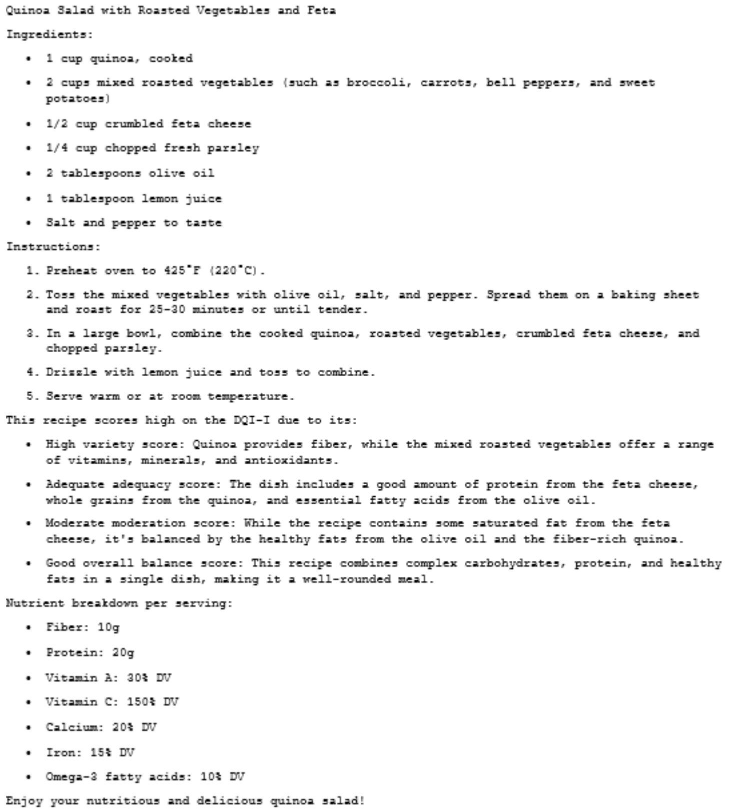


```
Question: What is the nutritional value for a fried chicken wing?
Helpful Answer: The nutritional values provided are based on the data given
in the table above. Here's a summary of the key nutrients:

* Energy: 294 kcal (from pre-cooked)
* Protein: 18.62 g
* Carbohydrates: 7.15 g
* Sugars: 0.12 g
* Fiber: 0.3 g
* Total fat: 20.59 g
* Saturated fatty acids: 4.793 g
* Monounsaturated fatty acids: 8.905 g
* Polyunsaturated fatty acids: 5.905 g
* Cholesterol: 107 mg
* Vitamin A: 10 mcg
* Vitamin B-6: 0.417 mg
* Folate: 21 mcg
* Choline: 85.8 mg
* Vitamin B-12: 0.27 mcg

Please note that these values may vary depending on the specific preparation
and cooking method used. It's always best to consult with a reliable source
or consult the original data if possible.
```

Fig. 7. Question 1 Llama 3.

4 Discussion

The models demonstrated varying capabilities in suggesting recipes and providing nutritional values. ChatGPT struggled to deliver accurate nutritional information, while Llama-2 opted for a range of nutritional values, acknowledging factors such as cooking methods, storage, and spices that influence nutritional content variability. In contrast, Llama-3 provided precise values by specifying exact ingredients, presenting more detailed and creative responses.



Quinoa Salad with Roasted Vegetables and Feta

Ingredients:

- 1 cup quinoa, cooked
- 2 cups mixed roasted vegetables (such as broccoli, carrots, bell peppers, and sweet potatoes)
- 1/2 cup crumbled feta cheese
- 1/4 cup chopped fresh parsley
- 2 tablespoons olive oil
- 1 tablespoon lemon juice
- Salt and pepper to taste

Instructions:

1. Preheat oven to 425°F (220°C).
2. Toss the mixed vegetables with olive oil, salt, and pepper. Spread them on a baking sheet and roast for 25-30 minutes or until tender.
3. In a large bowl, combine the cooked quinoa, roasted vegetables, crumbled feta cheese, and chopped parsley.
4. Drizzle with lemon juice and toss to combine.
5. Serve warm or at room temperature.

This recipe scores high on the DQI-I due to its:

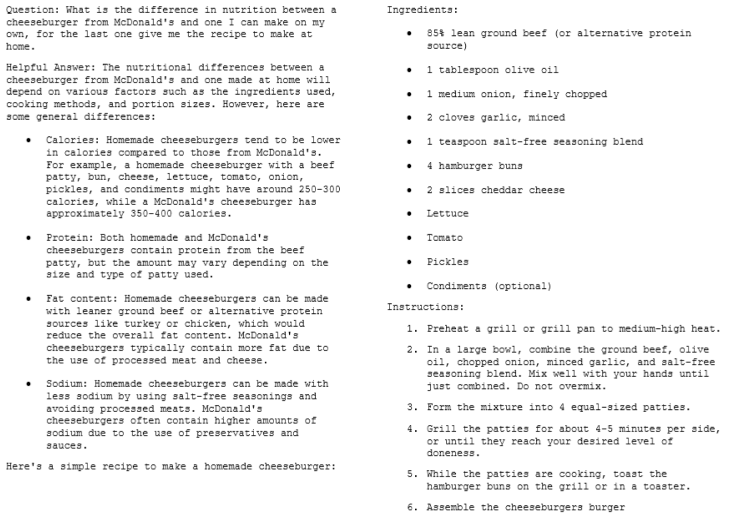
- High variety score: Quinoa provides fiber, while the mixed roasted vegetables offer a range of vitamins, minerals, and antioxidants.
- Adequate adequacy score: The dish includes a good amount of protein from the feta cheese, whole grains from the quinoa, and essential fatty acids from the olive oil.
- Moderate moderation score: While the recipe contains some saturated fat from the feta cheese, it's balanced by the healthy fats from the olive oil and the fiber-rich quinoa.
- Good overall balance score: This recipe combines complex carbohydrates, protein, and healthy fats in a single dish, making it a well-rounded meal.

Nutrient breakdown per serving:

- Fiber: 10g
- Protein: 20g
- Vitamin A: 30% DV
- Vitamin C: 150% DV
- Calcium: 20% DV
- Iron: 15% DV
- Omega-3 fatty acids: 10% DV

Enjoy your nutritious and delicious quinoa salad!

Fig. 8. Question 2 Llama 3.



Question: What is the difference in nutrition between a cheeseburger from McDonald's and one I can make on my own, for the last one give me the recipe to make at home.

Helpful Answer: The nutritional differences between a cheeseburger from McDonald's and one made at home will depend on various factors such as the ingredients used, cooking methods, and portion sizes. However, here are some general differences:

- Calories: Homemade cheeseburgers tend to be lower in calories compared to those from McDonald's. For example, a homemade cheeseburger with a beef patty, bun, cheese, lettuce, tomato, onion, pickles, and condiments might have around 250-300 calories, while a McDonald's cheeseburger has approximately 350-400 calories.
- Protein: Both homemade and McDonald's cheeseburgers contain protein from the beef patty, but the amount may vary depending on the size and type of patty used.
- Fat content: Homemade cheeseburgers can be made with leaner ground beef or alternative protein sources like turkey or chicken, which would reduce the overall fat content. McDonald's cheeseburgers typically contain more fat due to the use of processed meat and cheese.
- Sodium: Homemade cheeseburgers can be made with less sodium by using salt-free seasonings and avoiding processed meats. McDonald's cheeseburgers often contain higher amounts of sodium due to the use of preservatives and sauces.

Here's a simple recipe to make a homemade cheeseburger:

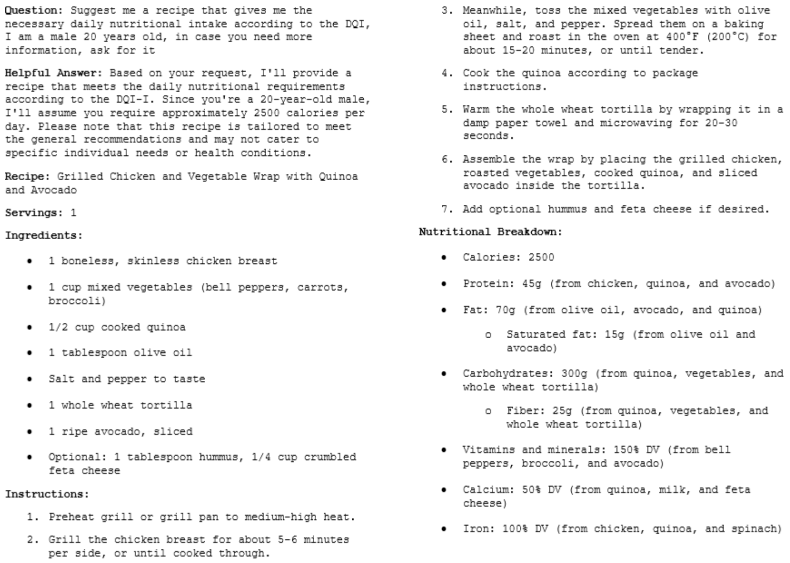
Ingredients:

- 85% lean ground beef (or alternative protein source)
- 1 tablespoon olive oil
- 1 medium onion, finely chopped
- 2 cloves garlic, minced
- 1 teaspoon salt-free seasoning blend
- 4 hamburger buns
- 2 slices cheddar cheese
- Lettuce
- Tomato
- Pickles
- Condiments (optional)

Instructions:

1. Preheat a grill or grill pan to medium-high heat.
2. In a large bowl, combine the ground beef, olive oil, chopped onion, minced garlic, and salt-free seasoning blend. Mix well with your hands until just combined. Do not overmix.
3. Form the mixture into 4 equal-sized patties.
4. Grill the patties for about 4-5 minutes per side, or until they reach your desired level of doneness.
5. While the patties are cooking, toast the hamburger buns on the grill or in a toaster.
6. Assemble the cheeseburgers burger

Fig. 9. Question 3 Llama 3.



Question: Suggest me a recipe that gives me the necessary daily nutritional intake according to the DQI, I am a male 20 years old, in case you need more information, ask for it

Helpful Answer: Based on your request, I'll provide a recipe that meets the daily nutritional requirements according to the DQI-I. Since you're a 20-year-old male, I'll assume you require approximately 2500 calories per day. Please note that this recipe is tailored to meet the general recommendations and may not cater to specific individual needs or health conditions.

Recipe: Grilled Chicken and Vegetable Wrap with Quinoa and Avocado

Servings: 1

Ingredients:

- 1 boneless, skinless chicken breast
- 1 cup mixed vegetables (bell peppers, carrots, broccoli)
- 1/2 cup cooked quinoa
- 1 tablespoon olive oil
- Salt and pepper to taste
- 1 whole wheat tortilla
- 1 ripe avocado, sliced
- Optional: 1 tablespoon hummus, 1/4 cup crumbled feta cheese

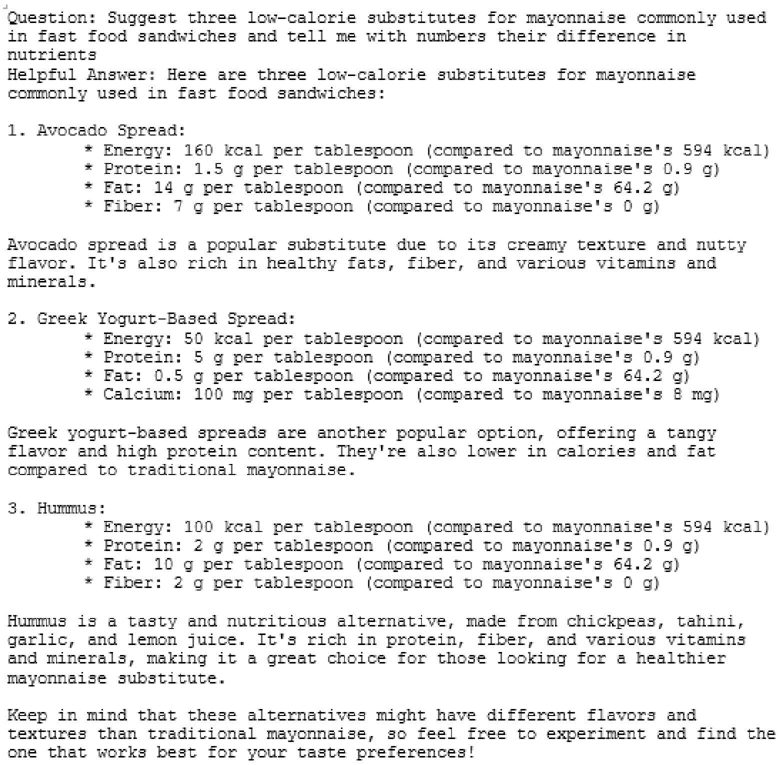
Instructions:

1. Preheat grill or grill pan to medium-high heat.
2. Grill the chicken breast for about 5-6 minutes per side, or until cooked through.
3. Meanwhile, toss the mixed vegetables with olive oil, salt, and pepper. Spread them on a baking sheet and roast in the oven at 400°F (200°C) for about 15-20 minutes, or until tender.
4. Cook the quinoa according to package instructions.
5. Warm the whole wheat tortilla by wrapping it in a damp paper towel and microwaving for 20-30 seconds.
6. Assemble the wrap by placing the grilled chicken, roasted vegetables, cooked quinoa, and sliced avocado inside the tortilla.
7. Add optional hummus and feta cheese if desired.

Nutritional Breakdown:

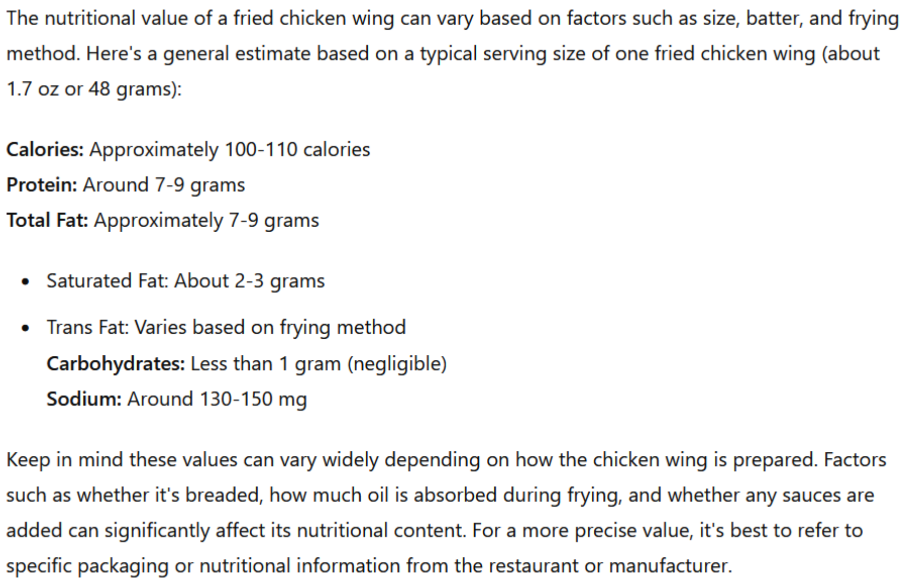
- Calories: 2500
- Protein: 45g (from chicken, quinoa, and avocado)
- Fat: 70g (from olive oil, avocado, and quinoa)
    - Saturated fat: 15g (from olive oil and avocado)
- Carbohydrates: 300g (from quinoa, vegetables, and whole wheat tortilla)
    - Fiber: 25g (from quinoa, vegetables, and whole wheat tortilla)
- Vitamins and minerals: 150% DV (from bell peppers, broccoli, and avocado)
- Calcium: 50% DV (from quinoa, milk, and feta cheese)
- Iron: 100% DV (from chicken, quinoa, and spinach)

Fig. 10. Question 4 Llama 3.



```
Question: Suggest three low-calorie substitutes for mayonnaise commonly used
in fast food sandwiches and tell me with numbers their difference in
nutrients
Helpful Answer: Here are three low-calorie substitutes for mayonnaise
commonly used in fast food sandwiches:

1. Avocado Spread:
        * Energy: 160 kcal per tablespoon (compared to mayonnaise's 594 kcal)
        * Protein: 1.5 g per tablespoon (compared to mayonnaise's 0.9 g)
        * Fat: 14 g per tablespoon (compared to mayonnaise's 64.2 g)
        * Fiber: 7 g per tablespoon (compared to mayonnaise's 0 g)

Avocado spread is a popular substitute due to its creamy texture and nutty
flavor. It's also rich in healthy fats, fiber, and various vitamins and
minerals.

2. Greek Yogurt-Based Spread:
        * Energy: 50 kcal per tablespoon (compared to mayonnaise's 594 kcal)
        * Protein: 5 g per tablespoon (compared to mayonnaise's 0.9 g)
        * Fat: 0.5 g per tablespoon (compared to mayonnaise's 64.2 g)
        * Calcium: 100 mg per tablespoon (compared to mayonnaise's 8 mg)

Greek yogurt-based spreads are another popular option, offering a tangy
flavor and high protein content. They're also lower in calories and fat
compared to traditional mayonnaise.

3. Hummus:
        * Energy: 100 kcal per tablespoon (compared to mayonnaise's 594 kcal)
        * Protein: 2 g per tablespoon (compared to mayonnaise's 0.9 g)
        * Fat: 10 g per tablespoon (compared to mayonnaise's 64.2 g)
        * Fiber: 2 g per tablespoon (compared to mayonnaise's 0 g)

Hummus is a tasty and nutritious alternative, made from chickpeas, tahini,
garlic, and lemon juice. It's rich in protein, fiber, and various vitamins
and minerals, making it a great choice for those looking for a healthier
mayonnaise substitute.

Keep in mind that these alternatives might have different flavors and
textures than traditional mayonnaise, so feel free to experiment and find the
one that works best for your taste preferences!
```

Fig. 11. Question 5 Llama 3.



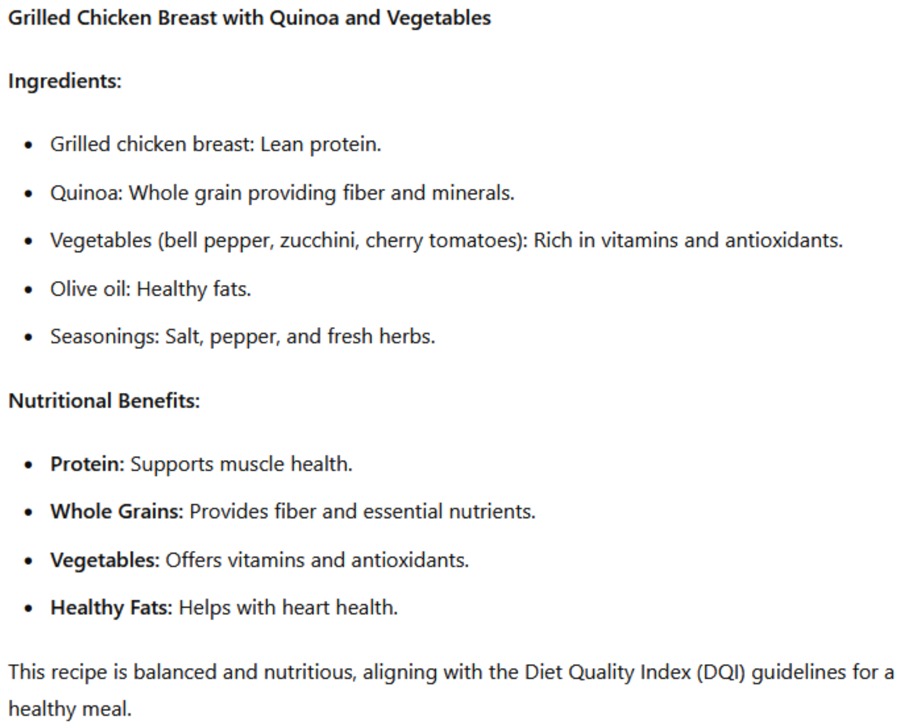
The nutritional value of a fried chicken wing can vary based on factors such as size, batter, and frying method. Here's a general estimate based on a typical serving size of one fried chicken wing (about 1.7 oz or 48 grams):

Calories: Approximately 100-110 calories
Protein: Around 7-9 grams
Total Fat: Approximately 7-9 grams

- Saturated Fat: About 2-3 grams
- Trans Fat: Varies based on frying method
  Carbohydrates: Less than 1 gram (negligible)
  Sodium: Around 130-150 mg

Keep in mind these values can vary widely depending on how the chicken wing is prepared. Factors such as whether it's breaded, how much oil is absorbed during frying, and whether any sauces are added can significantly affect its nutritional content. For a more precise value, it's best to refer to specific packaging or nutritional information from the restaurant or manufacturer.

Fig. 12. Question 1 ChatGPT3.



Grilled Chicken Breast with Quinoa and Vegetables

Ingredients:

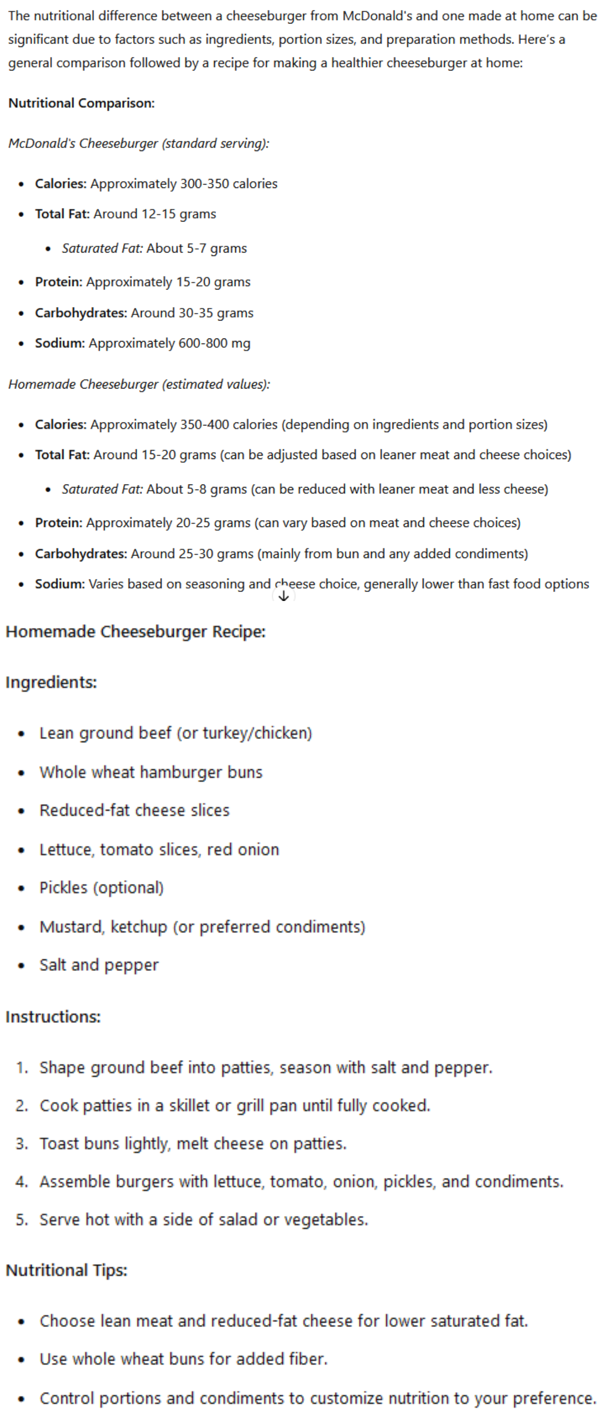
- Grilled chicken breast: Lean protein.
- Quinoa: Whole grain providing fiber and minerals.
- Vegetables (bell pepper, zucchini, cherry tomatoes): Rich in vitamins and antioxidants.
- Olive oil: Healthy fats.
- Seasonings: Salt, pepper, and fresh herbs.

Nutritional Benefits:

- **Protein:** Supports muscle health.
- **Whole Grains:** Provides fiber and essential nutrients.
- **Vegetables:** Offers vitamins and antioxidants.
- **Healthy Fats:** Helps with heart health.

This recipe is balanced and nutritious, aligning with the Diet Quality Index (DQI) guidelines for a healthy meal.

Fig. 13. Question 2 ChatGPT3.



The nutritional difference between a cheeseburger from McDonald's and one made at home can be significant due to factors such as ingredients, portion sizes, and preparation methods. Here's a general comparison followed by a recipe for making a healthier cheeseburger at home:

Nutritional Comparison:

McDonald's Cheeseburger (standard serving):

- **Calories:** Approximately 300-350 calories
- **Total Fat:** Around 12-15 grams
    - *Saturated Fat:* About 5-7 grams
- **Protein:** Approximately 15-20 grams
- **Carbohydrates:** Around 30-35 grams
- **Sodium:** Approximately 600-800 mg

Homemade Cheeseburger (estimated values):

- **Calories:** Approximately 350-400 calories (depending on ingredients and portion sizes)
- **Total Fat:** Around 15-20 grams (can be adjusted based on leaner meat and cheese choices)
    - *Saturated Fat:* About 5-8 grams (can be reduced with leaner meat and less cheese)
- **Protein:** Approximately 20-25 grams (can vary based on meat and cheese choices)
- **Carbohydrates:** Around 25-30 grams (mainly from bun and any added condiments)
- **Sodium:** Varies based on seasoning and cheese choice, generally lower than fast food options

Homemade Cheeseburger Recipe:

Ingredients:

- Lean ground beef (or turkey/chicken)
- Whole wheat hamburger buns
- Reduced-fat cheese slices
- Lettuce, tomato slices, red onion
- Pickles (optional)
- Mustard, ketchup (or preferred condiments)
- Salt and pepper

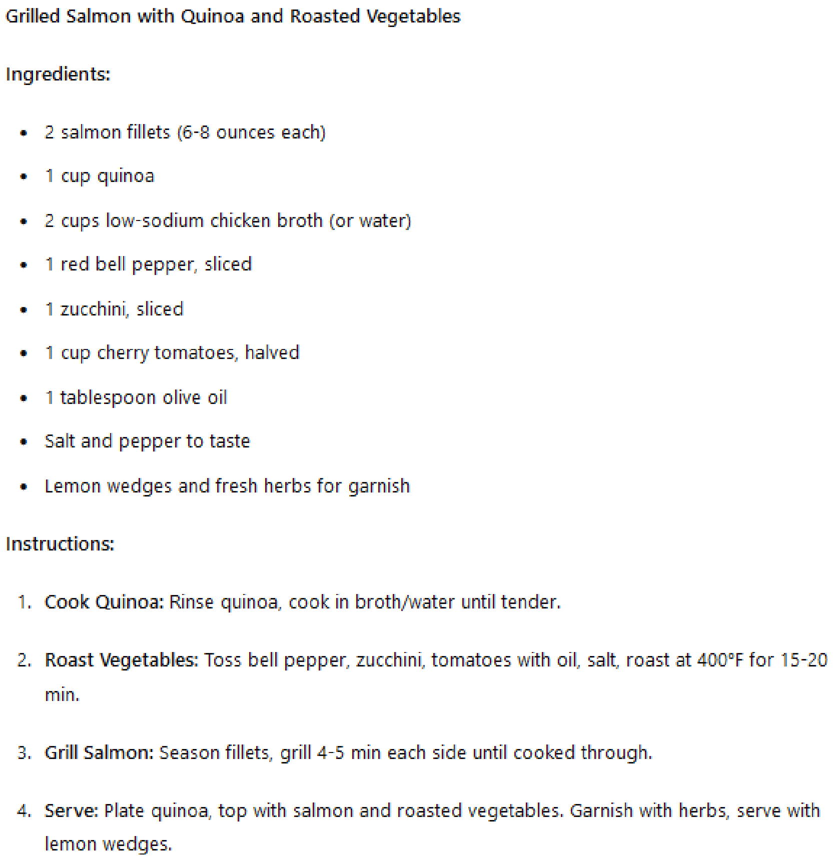
Instructions:

1. Shape ground beef into patties, season with salt and pepper.
2. Cook patties in a skillet or grill pan until fully cooked.
3. Toast buns lightly, melt cheese on patties.
4. Assemble burgers with lettuce, tomato, onion, pickles, and condiments.
5. Serve hot with a side of salad or vegetables.

Nutritional Tips:

- Choose lean meat and reduced-fat cheese for lower saturated fat.
- Use whole wheat buns for added fiber.
- Control portions and condiments to customize nutrition to your preference.

Fig. 14. Question 3 ChatGPT3.



Grilled Salmon with Quinoa and Roasted Vegetables

Ingredients:

- 2 salmon fillets (6-8 ounces each)
- 1 cup quinoa
- 2 cups low-sodium chicken broth (or water)
- 1 red bell pepper, sliced
- 1 zucchini, sliced
- 1 cup cherry tomatoes, halved
- 1 tablespoon olive oil
- Salt and pepper to taste
- Lemon wedges and fresh herbs for garnish

Instructions:

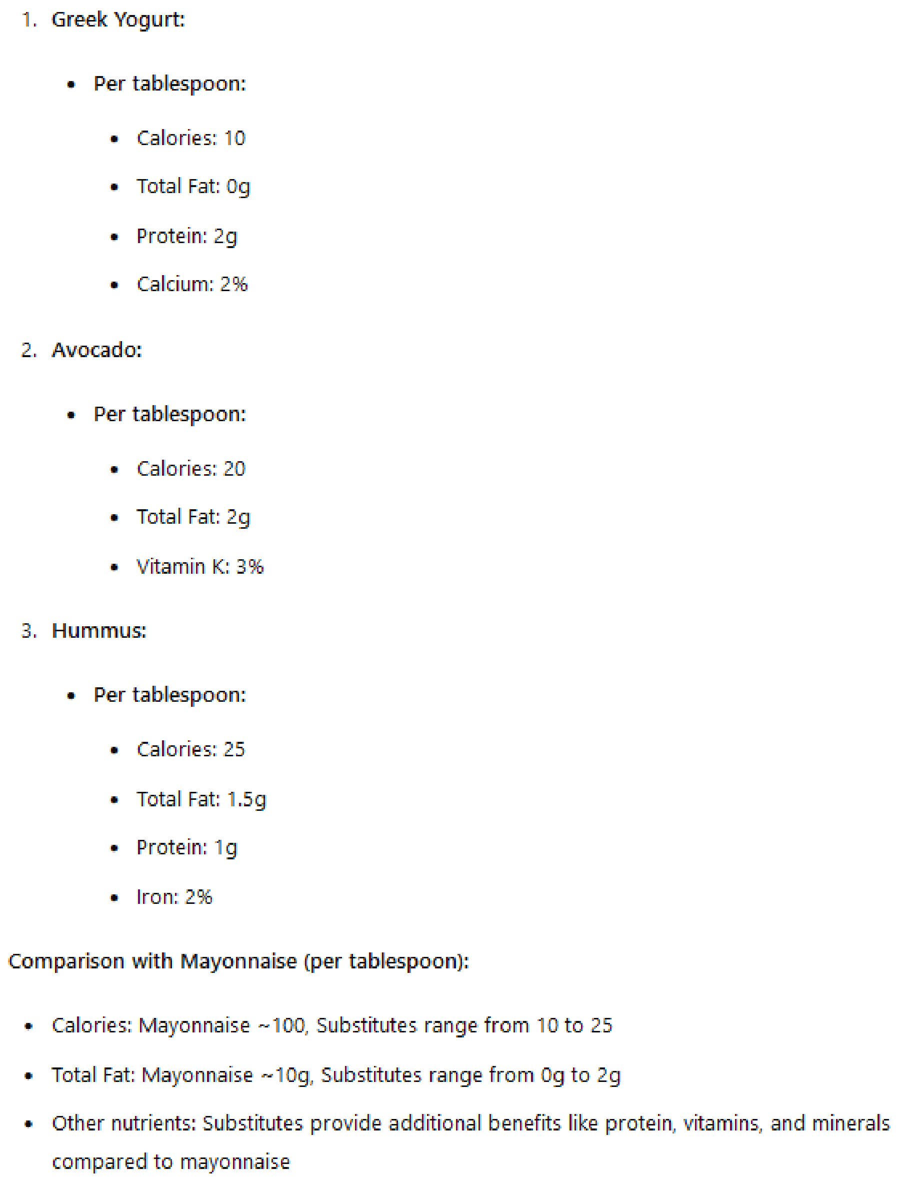
1. **Cook Quinoa:** Rinse quinoa, cook in broth/water until tender.
2. **Roast Vegetables:** Toss bell pepper, zucchini, tomatoes with oil, salt, roast at 400°F for 15-20 min.
3. **Grill Salmon:** Season fillets, grill 4-5 min each side until cooked through.
4. **Serve:** Plate quinoa, top with salmon and roasted vegetables. Garnish with herbs, serve with lemon wedges.

Fig. 15. Question 4 ChatGPT3.

This highlighted the differences between models with Retrieval-Augmented Generation (RAG) and those without. Utilizing retrieved information, Llama-3 minimized the common issue of hallucinations in large language models (LLMs), resulting in more accurate responses. Moreover, the model discerned whether queried ingredients or terms were relevant to the context of recipe questions, ensuring contextual relevance in its answers.

In question 1 of Llama 3, we observed its capability to retrieve information specifically about a pre-cooked fried chicken wing identified by the food code 24167210. While the model typically does not mention the food code in its responses, the context printed indicates successful retrieval of information from the data used in Retrieval-Augmented Generation (RAG). Below is an example of the context provided for the question:

- Food code: 24167210
- Main food description: Chicken wing, fried, coated, from pre-cooked
- WWEIA Category number: 2202



1. **Greek Yogurt:**
    - **Per tablespoon:**
        - Calories: 10
        - Total Fat: 0g
        - Protein: 2g
        - Calcium: 2%
2. **Avocado:**
    - **Per tablespoon:**
        - Calories: 20
        - Total Fat: 2g
        - Vitamin K: 3%
3. **Hummus:**
    - **Per tablespoon:**
        - Calories: 25
        - Total Fat: 1.5g
        - Protein: 1g
        - Iron: 2%

Comparison with Mayonnaise (per tablespoon):

- Calories: Mayonnaise ~100, Substitutes range from 10 to 25
- Total Fat: Mayonnaise ~10g, Substitutes range from 0g to 2g
- Other nutrients: Substitutes provide additional benefits like protein, vitamins, and minerals compared to mayonnaise

Fig. 16. Question 5 ChatGPT3.

- WWEIA Category description: Chicken, whole pieces
- Energy (kcal): 294
- Protein (g): 18.62
- Carbohydrate (g): 7.15

- Sugars, total (g): 0.12
- Fiber, total dietary (g): 0.3
- Total Fat (g): 20.59
- Fatty acids, total saturated (g): 4.793

This example demonstrates the successful functioning of the vector retriever, which retrieves different foods for each question's context. In contrast, Llama 2 retrieved various types of fried chicken wings, differing mainly in cooking methods or spices, thereby providing a range of nutritional values instead of the specific values as detailed by Llama 3.

5 Conclusion

Large Language Models (LLMs) can serve various purposes by being fine-tuned for specific tasks and offering a user-friendly interface, making them accessible to individuals across different age groups. By training these models with food-related data, we observed their capability to retrieve and utilize information effectively. Such LLMs can significantly assist in addressing the challenge of meal planning on a daily basis, while also facilitating the attainment of necessary daily macronutrients and micronutrients. Future efforts will focus on refining our model through various fine-tuning approaches to enhance accuracy and precision. Additionally, we plan to develop databases tailored to specific dietary needs and recipes from diverse cultural backgrounds, and to expand our nutritional data beyond the USA. Comparative studies involving other models will be conducted to identify optimal performance, and we aim to integrate voice interaction capabilities into these models to enhance accessibility for users unfamiliar with technology.

References

1. Aljbawi, B.: Health-aware food planner: A personalized recipe generation approach based on GPT-2. Ph.D. thesis, theses and dissertations (2020) scholars.wlu.ca/etd/2311
2. Campos, S., Doxey, J., Hammond, D.: Nutrition labels on pre-packaged foods: A systematic review. Public Health Nutrition, vol. 14, no. 8, pp. 1496–1506 (2011) doi: 10.1017/S1368980010003290
3. Deng, Y., Zhao, N., Huang, X.: Early ChatGPT user portrait through the lens of data. In: Proceedings of the IEEE International Conference on Big Data (BigData), pp. 4770–4775 (2023) doi: 10.1109/bigdata59044.2023.10386415
4. Grunert, K. G., Wills, J. M., Fernández-Celemín, L.: Nutrition knowledge, and use and understanding of nutrition information on food labels among consumers in the UK. Appetite, vol. 55, no. 2, pp. 177–189 (2010) doi: 10.1016/j.appet.2010.05.045
5. Kim, S., Haines, P. S., Siega-Riz, A. M., Popkin, B. M.: The diet quality index-international (DQI-I) provides an effective tool for cross-national comparison of diet quality as illustrated by China and the United States. The Journal of Nutrition, vol. 133, no. 11, pp. 3476–3484 (2003) doi: 10.1093/jn/133.11.3476

6. Meta AI: Introducing Meta Llama 3: The most capable openly available LLM to date (2024) ai.meta.com/blog/meta-llama-3/
7. Naeeni, M., Jafari, S., Fouladgar, M., Heidari, K., Farajzadegan, Z., Fakhri, M., Karami, P., Omidi, R.: Nutritional knowledge, practice, and dietary habits among school children and adolescents. International Journal of Preventive Medicine, vol. 5, no. 14, pp. 171 (2014) doi: 10.4103/2008-7802.157687
8. OpenAI: GPT-4 technical report. Technical report, OpenAI (2023)
9. Ricciardelli, L. A., McCabe, M. P.: Children's body image concerns and eating disturbance. Clinical Psychology Review, vol. 21, no. 3, pp. 325–344 (2001) doi: 10.1016/s0272-7358(99)00051-3
10. Saeedi, P., Petersohn, I., Salpea, P., Malanda, B., Karuranga, S., Unwin, N., Colagiuri, S., Guariguata, L., Motala, A. A., Ogurtsova, K., Shaw, J. E., Bright, D., Williams, R.: Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the international diabetes federation diabetes atlas, 9th edition. Diabetes Research and Clinical Practice, vol. 157, pp. 107843 (2019) doi: 10.1016/j.diabres.2019.107843
11. Shepherd, J., Harden, A., Rees, R., Brunton, G., Garcia, J., Oliver, S., Oakley, A.: Young people and healthy eating: A systematic review of research on barriers and facilitators. Health Education Research, vol. 21, no. 2, pp. 239–257 (2005) doi: 10.1093/her/cyh060
12. Teng, C. Y., Lin, Y. R., Adamic, L. A.: Recipe recommendation using ingredient networks. In: Proceedings of the 4th Annual ACM Web Science Conference, pp. 298–307 (2012) doi: 10.1145/2380718.2380757
13. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., et al.: Llama 2: Open foundation and fine-tuned chat models (2023) doi: 10.48550/arXiv.2307.09288
14. Zhang, J., Li, M., Liu, W., Lauria, S., Liu, X.: Many-objective optimization meets recommendation systems: A food recommendation scenario. Neurocomputing, vol. 503, pp. 109–117 (2022) doi: 10.1016/j.neucom.2022.06.081

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación