FILLET-A PLATFORM FOR INTELLIGENT NUTRITION

A Seminar Report

Submitted to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the degree

Bachelor of Technology

in

Computer Science and Engineering

by

SAPTHAMY P O

PTA19CS041

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING KALLOOPPARA

KERALA

DECEMBER 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING KALLOOPPARA

2022-23

CERTIFICATE

This is to certify that the seminar report entitled **Fillet-A platform for intelligent nutrition** submitted by **SAPTHAMY P O (**Reg. no. **PTA19CS041)** ,to the APJ Abdul Kalam Technological University in partial fulfilment of the B.Tech degree in Computer Science & Engineering is a bonafide record of the seminar work carried out by her under our supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Mrs. Talit Sara George	**Mrs. Jini George**	**Dr Renu George**
Coordinator	**Internal Guide**	**Head of the Department**
Dept. of CSE	Dept. of CSE	Dept. of CSE

ABSTRACT

Poor dietary behaviours are commonly associated with severe chronic diseases such as cardiovascular diseases, diabetes and obesity. Personalized food recommendation systems can be an important motivation to stimulate and inform people on best dietary practices by suggesting healthy foods and nutritionally balanced meals adjusted to their preferences and daily routines. The development of such systems require the process and integration of data available from different sources with different representations. FILLET is an intelligent platform for nutrition capable of collecting and integrating data from multiple sources including recipe websites, food blogs and nutrition databases. Components were developed for web scraping, identifying ingredients, estimating nutritional content and matching ingredients with food products from retailers to support a meal recommendation and shopping list assistance services. We present for each component the challenges identified in the literature and the ones we faced in their development, describing our approach and the lessons learned that can contribute to the future improvement of the platform and the development of related platforms.

ACKNOWLEDGMENT

I take this opportunity to express my deepest sense of gratitude and sincere thanks to everyone who helped me to complete this work successfully. I express my sincere thanks to **Dr.Renu George**, Head of Department, Computer Science and Engineering,College of Engineering Kallooppara for providing me with all the necessary facilities and support.

I would like to express my sincere gratitude to **Mrs.Talit Sara George** and **Mrs.Anitha Jose**, department of Computer Science and Engineering, College of Engineering Kallooppara for their support and co-operation. I would like to place on record my sincere gratitude to my seminar guide **Mrs.Jini George** Assistant Professor, Computer Science and Engineering, College of Engineering Kallooppara for the guidance and mentorship throughout the course .

Finally I thank my family, and friends who contributed to the successful fulfilment of this seminar work.

Sapthamy P O

CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 BACKGROUND

According to the World Health Organization, a healthy diet can help prevent several chronic diseases including obesity, diabetes and cardiovascular diseases. Personalized nutrition and food recommender systems are a trend to tackle this problem . These consist in automatic solutions to give personalized dietary advice considering individual user profiles to assist in acquiring healthier eating habits. Advice is usually provided in the form of individual food recommendations or as meal plans for a complete day or week . Relevant information to develop such systems is spread across different sources,such as recipe websites, foodie blogs and nutrition databases. Furthermore, the services provided by such systems can be extended to related areas such as health and exercise advice, as well as create additional business opportunities, including restaurant recommendation and promotion, automatic shopping list creation and fulfilment at retailers- especially important to meet increasing consumer demands regarding convenience in their shopping experience. These additional services extend the range of available data sources with relevant information. In general, data sources relevant for such a system are complex and diverse. They range from structured to highly unstructured databases, including some containing information with intermediate and mixed levels of structure. The challenges involved in collecting and organizing the data required by these systems may be one of the main reasons for their limited availability.

1.2 PURPOSE AND SCOPE

The purpose of fillet is collecting and integrating data from multiple, complex sources, the goal of the proposed platform is to provide tools for general data preparation operations that facilitate the development of additional functionalities to the services implemented with it. These methods include basic operations such

as web scraping and text pre-processing, as well as more advanced ones, such as named entity recognition. Currently, the platform includes components to support the development of a recipe recommendation and automated shopping list generation services. It uses data from recipe web sites, a nutritional food composition database, a product database and an ontology.

CHAPTER-2

LITERATURE REVIEW

"An Overview of Recommender Systems in the Healthy Food Domain"[1]

Trang Tran, T. N., Atas, M., Felfernig, A., & Stettinger, M. entitled "An Overview Of Recommendation Systems in the Healthy Food Doman" discusses using various machine approaches for recommendation systems such as collaborative filtering approach,content based filtering approach, knowledge based recommendation systems and hybrid recommendation system. All of these are recommendation systems for individuals.This paper also discuss about recommendation system for groups which means considering the preferences of every members in a group for building such a system the paper proposes aggregated model and aggregated predictions.

This paper also discusses a food recommendation system using deep learning. This system consists of two parts which are preference classification and food recommendation. For recipe classification, the system use Keras/tensorflow API to deployed DNN that contains multi-layer perceptron. For the food recommendation, developed the temporal model that evaluates eating history of a particular user to predict the next dishes to the user. This paper proposed a model to recommend dishes from user preference and eating history.The system contains a classification model to create a list of recommended dishes from user profile. Then create a temporal model that take list of dishes and eating history as input to increase the diversity of recommended dishes. To achieve the purpose, the system create a group of ingredients to increate accuracy of the model

“A Food Recommender System Considering Nutritional Information and User Preferences”[2]

Raciel Year Toledo ,Ahmad A. Alzahrani and Luis Martinez entitled “A Food Recommender Systems in the Healthy Food Domain” this paper introduces a system for meal recommendation based on nutritional content and user preference. The system is composed of four layers to process the information pipeline that begins in the user information layer and finishes in the final recommendation generation. The system capturing all the nutrition-related relevant information associated to the user. This information includes physiological data such as user height and weight, heart rate, burned calories, daily physical activity level; as well as information directly provided by the user such as daily food intake, and expert’s knowledge such as food composition tables and food’s exclusion criteria and also uses the sensorized Internet of Things (IoT) devices that allow a continuous information gathering in order to effectively build the user profile. The system uses a multicriteria decision analysis-based food pre-filtering approach for initially filtering out such foods which are not nutritionally appropriated to be recommended is proposed. An optimization-based menu recommendation model which contain three phases: the frequency-based menu generation, the probabilistic-based menu refining , and the restricted frequency-based menu generation ,which is used for recommending meals based on user preference and nutritional contents.

“A Machine Learning Approach to Recipe Text Processing”[3]

This paper propose a machine learning approach to recipe text processing problem aiming at converting a recipe text to a work flow. In this paper, they focus on the NLP such as word identification, named entity recognition, and syntactic analysis to extract predicate-argument structures (tuples of a verbal expression and its arguments) from a sentence in a recipe text. Predicate-argument structures are subgraphs of the work flow of a recipe. This paper is general and allows us to develop a text processing system in a certain domain very quickly with low cost.

“Extraction of Naming Concepts Based on Modifiers in Recipe Titles”[4]

Akiho Tachibana, Shoko Wakamiya, Hidetsugu Nanba and Kazutoshi Sumiya entitled “Extraction of Naming Concepts Based on Modifiers in Recipe Titles” , in this work, they proposed a method that extracts Naming Concepts for recipes, which are defined as characteristic elements summarized by modifiers in the recipes titles.Extracted different elements of ingredients and cooking utensils, determined the relations between them by calculating their degree of co-occurrence and extracted Naming Concepts by grouping the recipes based on feature patterns. The system identify Naming Concepts for the recipes by extracting feature patterns based on the differences extracted and grouping them on the basis of the patterns.

“Converting In-N-Out Orders into a Structured Form”[5]

This system is designed to convert written fast food orders from In-N-Out into a normal, structured form.The approach is to build off of PA3, using a Maximum Entropy Classifier for Named Entity Recognition, and then apply a PCFG Parser to convert the order into a normalized, tree form.The system also contains an additional stage consisting of rule-based conversion into a custom intermediate language existing between the NER stage and the parser stage in order to help with parsing.

“Information Extraction from Recipes”[6]

This project was to extract information from cooking recipes into a machine-interpretable format. In particular the project aims to identify which actions are applied to which ingredients, and possibly identify which utensils are being used. One potential application could be for an application that has to answer questions about the nature of the preparation of the food. Another, more far-fetched application could be to have a program learn correlations between ingredients, the actions that are applied to them, and the sequence of those actions in order to be able to novel recipes that are plausible. In this project, the goal is to reduce the preparation steps of a recipe to action, ingredient, utensil groups. The system

use a combination of NER and ideas from SRL in order to get this reduced form of the recipe.The system also uses an MEMM classifier, similar to that of PA3, to identify key words in the ingredient lists and preparation directions that are important for understanding the semantics of the recipe. The system uses SRL because of two reasons first, this approach would help convert recipes (and perhaps other kinds of instructions) to a more machine-interpretable form – a set of actions , each one associated with a set of ingredients and utensils, would give a computer or robot most of the information it needed to carry the instructions described in the recipe. Second, grouping ingredients with action and utensils could be useful in resolving coreference.

"Extracting Structured Data from Recipes Using Conditional Random Fields"[7]

This paper discusses a solution to solve structured prediction problem. This paper introduces discriminative structured prediction model called linear-chain conditional random field(CRF) ,which has been successful on similar tasks such as part-of-speech tagging and named entity recognition. The goal is to use data to learn a model that can predict the tag sequence for any ingredient phrase we throw at it, even if the model has never seen that ingredient phrase before. The beauty of the linear-chain CRF model is that it makes some conditional independence assumptions that allow us to use dynamic programming to efficiently search the space of all possible label sequences.

CHAPTER-3

TECHNOLOGIES

3.1 MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly. Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labelled and

unlabelled data for training – typically a small amount of labelled data and a large amount of unlabelled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labelled data requires skilled and relevant resources in order to train it or learn from it.

3.2 NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) can be defined asthe automatic (orsemi-automatic) processing of human language. The term 'NLP' issometimes used rather more narrowly than that, often excluding information retrieval and sometimes even excluding machine translation. NLP is sometimes contrasted with 'computational linguistics', with NLP being thought of as more applied. Nowadays, alternative terms are often preferred, like 'Language Technology' or 'Language Engineering'. Language is often used in contrast with speech (e.g., Speech and Language Technology). But I'm going to simply refer to NLP and use the term broadly. NLP is essentially multidisciplinary: it is closely related to linguistics (although the extent to which NLP overtly draws on linguistic theory varies considerably). It also has links to research in cognitive science, psychology, philosophy and maths (especially logic). Within CS, it relates to formal language theory, compiler techniques, theorem proving, machine learning and human-computer interaction. Of course it is also related to AI, though nowadays it's not generally thought of as part of AI.

3.3 DEEP NEURAL NETWORK

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases, and functions. These components functioning similar to the human brains and can be trained like any other ML algorithm.

DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features

from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network. For instance, it was proved that sparse multivariate polynomials are exponentially easier to approximate with DNNs than with shallow networks.

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level on its own.

Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain, learn from large amounts of data.. Deep learning allows machines to solve complex problems even when using a data set that is very diverse, unstructured and interconnected .

3.4 NAMED ENTITY RECOGNITION

Named Entity Recognition (NER) is a task in Information Extraction consisting in identifying and classifying just some types of information elements, called Named Entities (NE). As such it, serves as the basis for many other crucial areas in Information Management, such as Semantic Annotation, Question Answering, Ontology Population and Opinion Mining. The term Named Entity was first used at the 6th Message Understanding Conference (MUC), where it was clear the importance of the semantic identification of people, organizations and localizations, as well as numerical expressions such as time and quantities. Most of the NER tools nowadays keep considering these types of NE originated in MUC, though with important variations.

Semantic annotations go beyond familiar textual annotations about the contents of the documents to the formal identification of concepts and their relations. For example, a semantic annotation might relate the term Paris to an ontology identifying it as an instance of the abstract concept City, and linking it to the

instance France of the abstract concept Country, thus avoiding any ambiguity as to which Paris the text refers to. These annotations, intended primarily for their use by machines, bring two main benefits for these systems: enhanced information retrieval and improved interoperability. But then again, when applied on large collections, the automation is especially important in order to provide the scalability needed to annotate existing documents and reduce the burden of annotating new documents. This automation is typically implemented with Information Extraction techniques, among which Named Entity Recognition is used to identify concepts to annotate.

Question answering systems provide concrete answers to queries, and NER techniques are frequently used for this kind of systems as a means to facilitate the selection of answers. Indeed, in TREC-8 about 80% of the queries used to evaluate this kind of systems were of the type who, where and when , which use to be answered with named entities of type person, organization, localization and date.

The Semantic Web and all the applications it supports depend on technology to make information interoperable, and ontologies play a key role in this ambitious scenario . One of its cornerstones is therefore the proliferation of ontologies, which requires engineering for their quick and simple construction .One of the tasks in this line is the automatic population of ontologies, which aims at incorporating instances into existing ontologies without the intervention of humans. Named Entity Recognition emerges in this case to identify instances of the required concepts. An example of this kind of systems can be found in the tool KnowItAll , which applies bootstrapping techniques upon the Web to achieve this goal.

3.5 NEO4J

Neo4j stores and manages data in its more natural, connected state. The graph database takes a property graph approach, which is beneficial for both traversal performance and operations runtime. Neo4j has evolved into a rich ecosystem with many tools, applications, and libraries, which give you opportunity to integrate graph technologies with your working environment. Neo4j is the world’s leading graph database. The architecture is designed for optimal management, storage, and traversal of nodes *and* relationships. Neo4j Aura is

the fully managed graph data platform service in the cloud. Aura empowers developers and data scientists to quickly build scalable, AI-driven applications and analyze big data with algorithms without the hassle of managing infrastructure. Neo4j Aura includes AuraDB for applications and AuraDS for data science. Neo4j Graph Data Science is a connected data analytics and machine learning platform that helps you understand the connections in big data. Cypher is Neo4j's graph query language that allows users to store and retrieve data from the graph database. It is a declarative, SQL-inspired language for describing patterns in graphs. The syntax provides a visual and logical way to match patterns of nodes and relationships in the graph. Cypher has been designed to be easy to learn, understand, and use for everyone, but also incorporate the power and functionality of other standard data access languages.

CHAPTER-4

METHODOLOGY

4.1 VITERBI ALGORITHM

The Viterbi algorithm is a dynamic programming algorithm for obtaining the maximum a posteriori probability estimate of the most likely sequence of hidden states—called the Viterbi path—that results in a sequence of observed events, especially in the context of Markov information sources and hidden Markov models (HMM).

The algorithm has found universal application in decoding the convolutional codes used in both CDMA and GSM digital cellular, dial-up modems, satellite, deep-space communications, and 802.11 wireless LANs. It is now also commonly used in speech recognition, speech synthesis, diarization, keyword spotting, computational linguistics, and bioinformatics. For example, in speech-to-text (speech recognition), the acoustic signal is treated as the observed sequence of events, and a string of text is considered to be the "hidden cause" of the acoustic signal. The Viterbi algorithm finds the most likely string of text given the acoustic signal.

The Viterbi algorithm is named after Andrew Viterbi, who proposed it in 1967 algorithm for convolutional codes over noisy digital communication links. It has, however, a history of multiple invention, with at least seven independent discoveries, including those by Viterbi, Needleman and Wunsch, and Wagner and Fischer. It was introduced to Natural Language Processing as a method of part-of-speech tagging as early as 1987.

Viterbi path and Viterbi algorithm have become standard terms for the application of dynamic programming algorithms to maximization problems involving probabilities.[3] For example, in statistical parsing a dynamic programming algorithm can be used to discover the single most likely context-free derivation (parse) of a string, which is commonly called the "Viterbi parse". Another application is in target tracking, where the track is computed that assigns a maximum likelihood to a sequence of observations.

A generalization of the Viterbi algorithm, termed the max-sum algorithm (or max-product algorithm) can be used to find the most likely assignment of all or some subset of latent variables in a large number of graphical models, e.g. Bayesian networks, Markov random fields and conditional random fields. The latent variables need, in general, to be connected in a way somewhat similar to a hidden Markov model (HMM), with a limited number of connections between variables and some type of linear structure among the variables. The general algorithm involves message passing and is substantially similar to the belief propagation algorithm (which is the generalization of the forward-backward algorithm).

With the algorithm called iterative Viterbi decoding one can find the subsequence of an observation that matches best (on average) to a given hidden Markov model. This algorithm is proposed by Qi Wang et al. to deal with turbo code. Iterative Viterbi decoding works by iteratively invoking a modified Viterbi algorithm, reestimating the score for a filler until convergence.

An alternative algorithm, the Lazy Viterbi algorithm, has been proposed. For many applications of practical interest, under reasonable noise conditions, the lazy decoder (using Lazy Viterbi algorithm) is much faster than the original Viterbi decoder (using Viterbi algorithm). While the original Viterbi algorithm calculates every node in the trellis of possible outcomes, the Lazy Viterbi algorithm maintains a prioritized list of nodes to evaluate in order, and the number of calculations required is typically fewer (and never more) than the ordinary Viterbi algorithm for the same result. However, it is not so easy to parallelize in hardware.

4.2 CONDITIONAL RANDOM FIELD MODEL

Conditional random fields (CRFs) are a class of statistical modelling methods often applied in pattern recognition and machine learning and used for structured prediction. Whereas a classifier predicts a label for a single sample without considering "neighbouring" samples, a CRF can take context into account. To do so, the predictions are modelled as a graphical model, which represents the presence of dependencies between the predictions. What kind of graph is used depends on the application. For example, in natural language processing, "linear chain" CRFs are popular, for which each prediction is dependent only on its immediate neighbours. In image processing,

the graph typically connects locations to nearby and/or similar locations to enforce that they receive similar predictions.

Other examples where CRFs are used are: labelling n or parsing of sequential data for natural language processing or biological sequences, part-of-speech tagging, shallow parsing, named entity recognition, gene finding, peptide critical functional region finding, and object recognition and image segmentation in computer vision.

CRFs can be extended into higher order models by making each dependent on a fixed number of previous variables . In conventional formulations of higher order CRFs, training and inference are only practical for small values of (such as $k \leq 5$), since their computational cost increases exponentially with k.

However, another recent advance has managed to ameliorate these issues by leveraging concepts and tools from the field of Bayesian nonparametric. Specifically, the CRF-infinity approach constitutes a CRF-type model that is capable of learning infinitely-long temporal dynamics in a scalable fashion. This is effected by introducing a novel potential function for CRFs that is based on the Sequence Memorizer (SM), a nonparametric Bayesian model for learning infinitely-long dynamics in sequential observations. To render such a model computationally tractable, CRF-infinity employs a approximation of the postulated novel potential functions (which are driven by an SM). This allows for devising efficient approximate training and inference algorithms for the model, without undermining its capability to capture and model temporal dependencies of arbitrary length.

There exists another generalization of CRFs, the semi-Markov conditional random field (semi-CRF), which models variable-length segmentations of the label sequence .This provides much of the power of higher-order CRFs to model long-range dependencies of the , at a reasonable computational cost.

Finally, large-margin models for structured prediction, such as the structured Support Vector Machine can be seen as an alternative training procedure to CRFs.

4.3 LANGUA ALIMENTARIA THESAURUS

Clear, unambiguous food descriptions are essential to enable users to correctly identify and select foods required from a food composition database (FCDB) and to facilitate interchange of food composition data. EuroFIR has established a

common standard for the identification and description of foods in European FCDBs that allows for application of state-of-the-art concepts in database linking and management and their comparability as well as the comparison and interchange of food composition data. The food description system chosen was LanguaL (Langua aLimentaria or language of food), an international controlled vocabulary for systematic food description. EuroFIR has supported new versions of the LanguaL thesaurus, including the 2008 version

In addition, prototype food description software, LanguaL Food Product Indexer, was developed to help FCDB compilers index the foods in their databases, and hence to allow record retrieval on the EuroFIR platform. Having attended specialised training courses on indexing using LanguaL, national compilers in the EuroFIR network have been undertaking the enormous task of indexing their food composition datasets. Over 29 European datasets have been indexed, covering over 29,000 foods. A number of specialised datasets, including EuroFIR-BASIS, and other datasets, such as the USDA dataset, have also been indexed. Indexing has been subject to quality assessment, taking account of both reproducibility and correctness. Compilers have also helped to improve the LanguaL thesaurus by providing feedback and by translating food names to local languages.

4.4AGROVOOC

The Food and Agriculture Organization of the United Nations (FAO) has coordinated AGROVOC, a valuable tool for data to be classified homogeneously, facilitating interoperability and reuse. AGROVOC is a multilingual and controlled vocabulary designed to cover concepts and terminology under FAO's areas of interest. It is a relevant Linked Open Data set about agriculture, available for public use, and its highest impact is through facilitating the access and visibility of data across domains and languages. AGROVOC provides a way to organize knowledge for subsequent data retrieval. It is a structured collection of concepts, terms, definitions and relationships. Concepts represent anything in food and agriculture, such as maize, hunger, aquaculture, value chains or forestry. These concepts are used to unambiguously identify resources, allowing standardized indexing processes, making searching more efficient. Each concept in AGROVOC also has terms used to express it in

various languages, so called lexicalizations. Today, AGROVOC consists of +40 600 concepts and +963 000 terms in up to 41 languages. AGROVOC is a relevant thesaurus about food and agriculture, published as linked open data, available for public use.

CHAPTER-5

ARCHITECTURE

The platform architecture is represented below and relies on a three layer process.

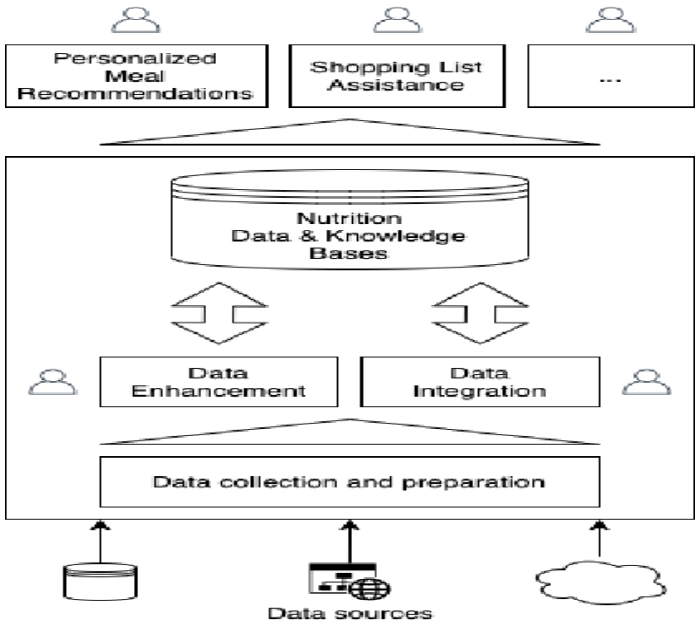

Fig 5.1 Architecture of fillet

The first layer, data preparation, concerns the low level access to the information source and its consolidation. Data may be collected using different methods including web services (either exposing or invoking), database queries and scrapers. Its implementation is mostly specific to each source and, in this step, data is also normalized and adapted to the platform's internal structure (holding semi-structured data at this point). In the next layer, data enhancement, we use models and algorithms to discover additional knowledge about the data being integrated. Named entity recognition and sentiment analysis are examples of operations that can provide richer information about the content being integrated. Data integration, the last layer, uses the information collected in the previous layers to associate information between different sources. Identifying common recipes from different websites, correlating ingredients in reviews with respective information from food composition databases or recipe ingredients with food products are examples of operations to be performed in this phase. Users are involved in the data enhancement and integration layers to validate the information and ensure data quality. This formal process facilitates the platform's extension with new features and adaptation to new sources of information. The platform exposes integration web services to allow accessing all functionalities including any

discovered associations and information. This allows developing different services using FILLET's created knowledge.

The system consist of modules to support the development of a personalized meal recommender and shopping list creation services . The objective of this implementation is to integrate recipes from websites with food composition databases and food products. Modules for nutrition and shopping assistance Fig. 3. Recipe scraper module from retailers. The result consists in richer information of each recipe including an estimation of its nutritional content, also including an implicit link to the LanguaL™ thesaurus, and an association to the necessary food products from retailers' product databases. This information is important for the development of personalized nutrition for allowing to consider recipes' nutritional content as well as user's restrictions and preferences. LanguaL™ contributes to making this possible by allowing, for instance, to distinguish meat from vegetables. Knowing the necessary products to prepare a given recipe also allows estimating its price and to facilitate shopping assistance services. In the following subsections we'll describe the modules that have been implemented in this platform: 1) the scraper belonging to the data collection and preparation layer, 2) the ingredient line parser from the data enhancement layers, 3) the nutrition content estimator, and 4) food product matching modules from the data integration layer.

CHAPTER-6

MODULES

6.1 RECIPE SCRAPER

The web is a source of a lot of widely available information that is relevant for nutrition. This includes recipes, foodie blogs, reviews (of recipes, restaurants, products...), etc. Web scraping is a commonly used technique to extract information directly from web pages. We implemented a recipe scraper module for FILLET, capable of extracting recipe information from different websites in order to easily collect a big amount of diversified recipes. The main challenges related to scraping recipes from websites are related to the different structures used by websites and the different information that is contained. This is caused by the lack of a widely adopted standard structure to represent recipes on websites. We used Scrapy4 to base our implementation of the recipe website scraper and opted for a modular architecture to be able to adapt to each website . Adapters are developed specifically for each website and are responsible for scraping the information and normalizing it into a common recipe structure. For the current version, we decided to focus on four Portuguese recipe websites (yammi.pt, chef.continente.pt, teleculinaria.pt, and receitasportuguesas.com). The normalized recipe structure represents the most common information included in websites including title, ingredient list, preparation steps, difficulty and preparation time. The chosen architecture facilitates the extension for new recipe websites and also enables the support for other kinds of data structures and sources such as reviews or comments. The recipe scraper is the first step for structuring recipe information in the FILLET platform.

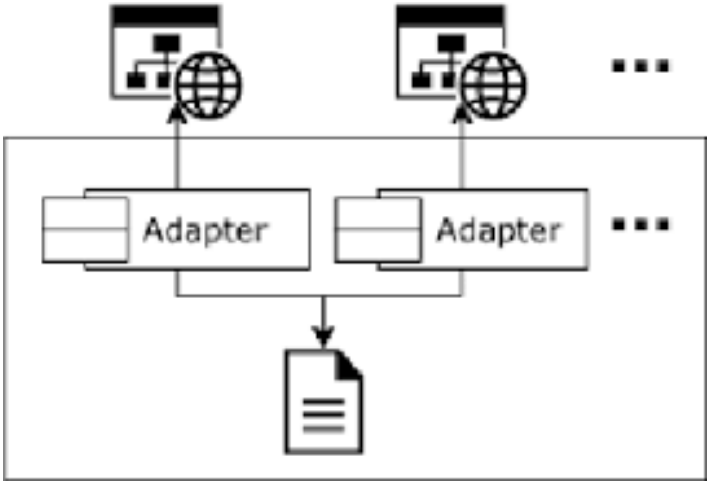

Fig 6.1 Recipe Scraper

6.2 INGREDIENT LINE PARSER

The ingredient line parser module belongs to the nutrition specific data preparation layer which divides ingredient lines into quantities, units, name and other information. This additional information can either be preparation notes or specific ingredient details. This module is a cornerstone of any nutrition application using recipes, as it collects the ingredients estimation and the food product matching modules. The ingredient parsing problem can be formulated as an information extraction problem, but it is very specific as ingredient line formats generally follow repetitive structures. Three sentences are depicted as illustrations of these repetitive structures along with their respective named entities. The complexity of this problem is hidden in exceptions to this pattern.

In FILLET, they started by using the NY Team's CRF model , adapting the feature set to work with recipes written in Portuguese and its structure. This also included definition of rules for the n-grams, namely unigrams and bigrams. The training data was extracted from a group of Portuguese websites which was annotated using an automated rule-based system followed by a manual validation. A preliminary version of this work was already published, using a corpus of English recipes. Using this Portuguese recipe dataset with algorithm improvements, an F1-score of 0.91 was achieved. Limitations of this approach are mixtures of units, nonnumeric quantities, and mixtures of ingredients and food processing and ingredient details. With the premise of structure uniformity between ingredient lines, we decided to implement a simpler methodology for comparison. This baseline approach relied on RE rules, following a similar strategy implemented by Christensen . It used a set of grammatical

templates that included the most common ingredient line structures, in which regular expression rules would loop through for extracting each entity. Not matched exceptions were stored for analysis allowing the creation of additional rules. It achieved an F1-score of 0.74, lower then the CRF model. The reason behind this result is mainly due to ingredient line structures that fall outside the standard and, therefore, require numerous additions to the rule and template set. In opposition to Christensen approach, we believe that the model should be dynamic and able to learn to predict variable ingredient line structures. In other words, the continuous addition of new rules to the model is costly and not scalable. Therefore, the best strategy is to build, on top of this rule-based logic, a learning approach capable of understanding the exceptions.

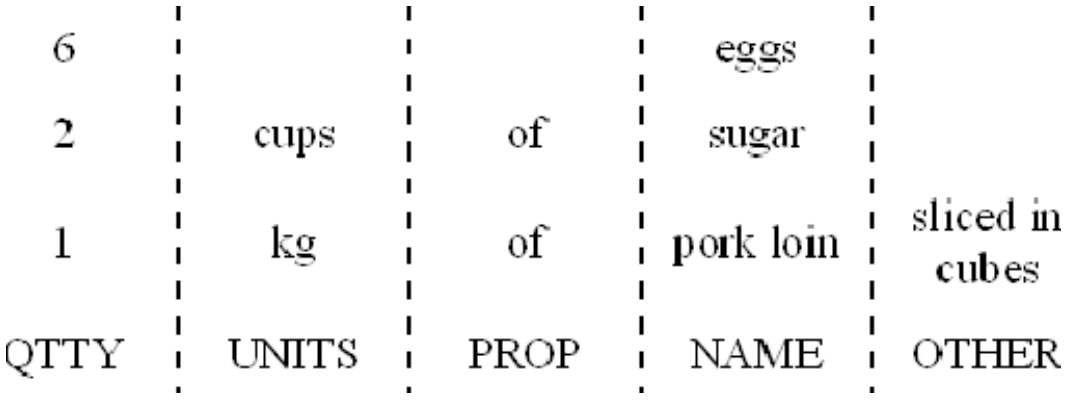

Fig 6.2 Ingredient Line Parser

6.3 NUTRITIONAL CONTENT ESTIMATION

The Nutritional Content Estimation module allows for the automatic processing of text of a culinary recipe to estimate its nutritional value, obtaining predictions for energy, macro-, and micro-nutrient content of a given dish. This information can then be used to more adequately formulate a weekly meal plan, according to a user's nutritional needs and preferences.

This module performs nutritional value estimation of recipes by applying Information Extraction techniques on the available recipe text, obtaining data regarding composing ingredients and how they are cooked, associating these to already established national FCDBs, and interpolating the final overall quantity of macro and micro-nutrients

present in a given dish. For the development dataset of Portuguese recipes, the TabeladeComposicao de Alimentos ˜ (TCA) from INSA was used, containing nutritional information for 962 commonly used ingredients and dish components in Portuguese cuisine. Each entry in this FCDB has its composition in terms of macro and micro nutrients identified, as well as its corresponding LanguaL™ annotation. The main ingredients of a given recipe are identified and extracted as described in Subsection II-B. To identify how each ingredient is cooked, this module analyses the preparation steps of a recipe, extracting cooking related verbs with a trained ML classifier, and associating direct/indirect objects of said verbs that contain ingredient names with a LanguaL™ code that pertains to cooking methods (G facet), as schematized . This classifier identifies cooking verbs and differentiates them into three main cooking methodology categories, corresponding to a first level G-facet LanguaL™ code: cooked by dry heat, cooked by moist heat, and cooked with fat or oil. These categories were chosen since they group the most prevalent cooking methods in the dataset, and they are the most useful to distinguish the nutritional values of the same ingredient cooked in different ways in the TCA. The identification of a verb's direct/indirect objects is obtained by predicting the dependency tree of each sentence; dependency parsing is performed using a pretrained parser for Portuguese.10 The classifier, based on extremely random decision trees, was trained with 26889 words extracted from the instruction steps of 333 Portuguese recipes, obtained from web scraping, and annotated manually. Words were classified independently

to identify which of them pertain to cooking method. The Portuguese fast Text word embeddings for each word were used as a 300-dimensional input for classifier training, with 5 fold cross-validation, and 4 target classes (no cooking method, cooked with dry heat, cooked with moist heat, and cooked with fat). 8743 words from 112 cooking recipes were used for testing, achieving a micro-average F1 score of 0.91 on a word level (micro-average precision of 0.95 and micro-average recall of 0.88). For each recipe, the aforementioned methodologies of cooking method identification and association with ingredients is applied, along with text based queries on ingredient names present in the TCA, to produce a list of the most probable ingredients of a given recipe and how each is cooked. To test the efficacy of this module, 100 web-scraped recipes had their ingredients manually annotated with entries from the TCA, with adequate variations according to the cooking methodologies used along the recipe

(when available). The predicted list of ingredients was compared with the annotation for each recipe: a mean example-based accuracy of 0.39 was achieved, with a mean example-based F1-score of 0.54. Although these results can be improved upon, they show promising capabilities for decomposing a recipe into its constituent ingredients, bearing in mind how these are cooked. With this, associations with FCDBs can be made, resulting in adequate nutritional value estimation of a given dish. During the development of the cooking methods classifier, it was noted that the overall performance on FCDB ingredient retrieval was only slightly improved, when compared with the more accurate iterations of the classifier. This indicates that the correct identification of cooking methods in recipe instructions is not the major limiting factor. Two possibilities for further improvement arise: ingredient to cooking method association, and text search of ingredients on the TCA. The association between an identified cooking method and the ingredients that are subjected to it is currently performed by analysing the dependency tree of the cooking verb. This does not guarantee a correct ingredient association if, for instance, dependency parsing fails, the ingredient is referred in a previous sentence, or if the ingredient is part of a mixture of ingredients previously defined in the instructions. Thus, this association strategy can be improved as future work, using more robust argument predicate analysis and representation, as seen in Kiddon et al.. In addition, to identify an ingredient on the TCA, a text based search is performed on the names of all the registered ingredients, which often leads to irrelevant entries to be considered as well. Filtering out these outliers remains a challenge in this module. Currently, priority is given to ingredients that have the least components added, as represented in the number of associated H facet LanguaL™ codes, but more restrictions can be applied in the future to improve results.

The approach, is that the model should be dynamic and able to learn to predict variable ingredient line structures. In other words, the continuous addition of new rules to the model is costly and not scalable. Therefore, the best strategy is to build, on top of this rule-based logic, a learning approach capable of understanding the exceptions.

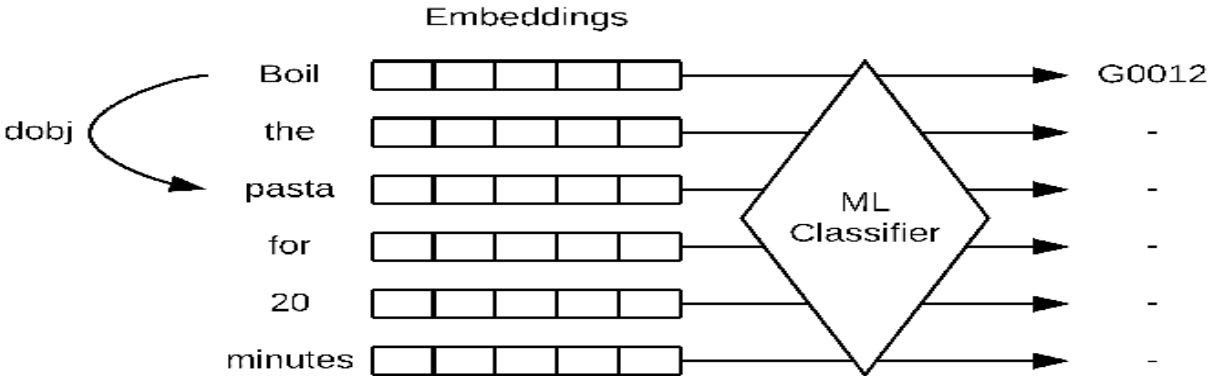

Fig 6.3 Nutritional content estimation using ml classifier

6.4 FOOD PRODUCT MATCHING

The food product matching module aims at associating ingredients identified in recipes with suitable food products from retailers. This information is important for the development of personalized meal recommendation and shopping assistance services for inferring the necessary products to prepare recipes and estimating its cost.

Thus paper have based the implementation in the Neo4j Graph database to store all the information of the system. The product information from retailers.Product matching simplified data model ingredients identified in recipes and the AGROVOC thesaurus were loaded to the database .AGROVOC is a multi-lingual thesaurus containing relevant vocabulary from the food domain. For each concept it contains multiple labels in different languages and alternative nomenclatures. When loading the information to the database, all products and ingredients are associated with the most similar AGROVOC concept, according to Neo4j's full text search index on AGROVOC Labels. We developed a machine learning based approach using a dataset previously created for a different purpose. In this dataset, the ingredients were not parsed from recipes but created manually. A preliminary approach was developed to tackle this problem using two steps: generating candidates consisting of pairs of candidate matches followed by a binary classifier to determine if each pair is a positive match. The candidate generation is bidirectional, i.e., we may generate candidates for new products or for new ingredients. We combine three data model paths to create candidate pairs: 1) direct matching between ingredient and product names, 2) matching between ingredient names and all product categories in the hierarchy, and 3) using AGROVOC, limiting the distance between common AGROVOC concepts. The number of candidates is limited to the best matches to control the number of final elements to be

processed in the next step. In 97% of the cases, the positive match is included in the generated set. In the next step, we compute features for each potential match. The features consist in the distance between AGROVOC concepts and different complementary text distance functions between candidate names, namely the hamming, cosine and longest common substring similarity (lcsstr) functions from the textdistance python package. A SVM binary classifier was then trained to identify positive and negative matches. We used group aware shuffle splitting to divide the dataset into training (80%) and test (20%) to ensure all candidates for a given product end up in the same split. The validation with the test set resulted in a macro average F1-score of 0.82. Since there may be multiple positive matches for the same candidate, we filtered the best according to the classifier's certainty, achieving a macro average F1-score of 0.93. The model is used in the FILLET platform to create match suggestions to be validated by the user. This process is triggered every time new products or ingredients are added to the system. A graphical user interface is available to allow reviewing the suggestions and validating the associations.

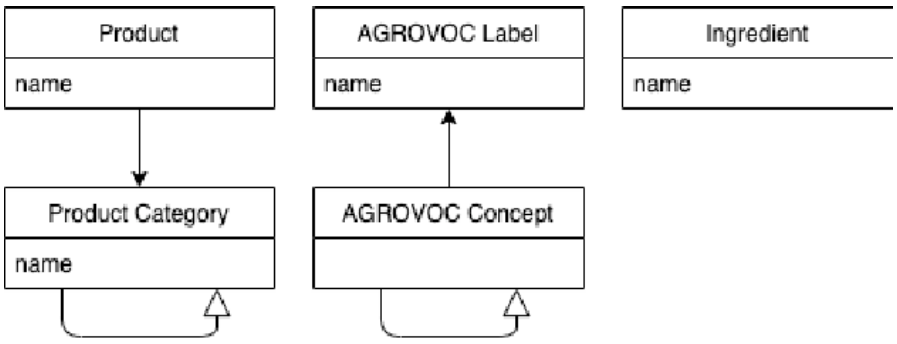

Fig 6.4 Food product matching

6.5 RECOMMENDATION MODULE

Collaborative filtering recommender systems (CF)

Collaborative filtering recommender systems (CF) CF became one of the most researched techniques of recommender systems. The basic idea of CF is to use the wisdom of the crowd for making recommendations. First of all, a user rates

some given items in an implicit or explicit fashion. Then, the recommender identifies the nearest neighbors whose tastes are similar to those of a given user and recommends items that the nearest neighbors have liked ,CF is usually implemented on the basis of the following approaches: user-based ,item-based ,model-based approaches and matrix factorization .

Content-based recommender systems (CB)

Content-based recommender systems (CB) These systems can make a personalized recommendation by exploiting information about available item descriptions (e.g., genre and director of movies) and user profiles describing what the users like. The main task of a CB system is to analyze the information regarding user preferences and item descriptions consumed by the user, and then recommend items based on this information. Knowledge-based recommender systems (KBS) KBS are recognized as a solution for tackling some problems generated by classical approaches (e.g., ramp-up problems .Moreover, these systems are especially useful in domains where the number of available item ratings is very low (e.g., apartments, financial services) or when users want to define their requirements explicitly (e.g., “the color of the car should be white”). There are two main approaches for developing knowledge-based recommender systems: case-based recommendation and constraint-based recommendation addition, critiquing-based recommendation is considered as a variant of case-based recommendation. This approach uses users’ preferences to recommend specific items, and then elicits users’ feedback in the form of critiques for the purpose of improving the recommendation accuracy.

Hybrid recommender systems (HRS) HRS are based on the combination of the techniques. According to Ricci et al. “A hybrid system combining techniques A and B tries to use the advantages of A to fix the disadvantages of B”. For instance, CF methods have to face the new-item problem. Whereas, CB approaches can tackle this problem because the prediction for new items is usually based on available descriptions of these items. Burke presents some hybrid approaches which combine both CF and CB, including weighted, switching, mixed, feature combination, cascade, feature augmentation, and meta-level.

CHAPTER-7

CONCLUSION

FILLET, a platform designed to facilitate the extraction, structuring and integration of information from heterogeneous sources of the food domain: both structured and unstructured. Its architecture relies on three layers to collect, prepare, enhancer and integrate information from the different sources. It also describe components developed for different FILLET layers including their related work, associated challenges, our methodology and preliminary results. FILLET is envisioned as a generic platform to support the development of different services related to food and nutrition including personalized meal recommendations and shopping list assistance, creating new business opportunities for retailers, restaurants among other entities. The experience shared in this work contributes to the development of related platforms and services. In the future, the developed modules will be improved by annotating additional data, testing different methodologies and will be subjected to further validation. Additional modules will also be developed towards FILLET's vision.

REFERENCES

[1] Thi Ngoc Trang Tran et al. “An Overview of Recommender Systems in the Healthy Food Domain”. en. In: Journal of Intelligent Information Systems 50.3 June 2018

[2] Raciel Yera Toledo, Ahmad A. Alzahrani, and Luis Mart´ınez. “A Food Recommender System Considering Nutritional Information and User Preferences”. 2012

[3] Shinsuke Mori et al. “A Machine Learning Approach to Recipe Text Processing”.In:Ptoceedings of the 1st Cooking With Computer Workshop.2012

[4] Akiho Tachibana et al. “Extraction of Naming Concepts Based on Modifiers in Recipe Titles”. In: Lecture Notes in Engineering and Computer Science 2209 (Mar. 2014), pp. 507–512. [5] Anders Møller and Jayne Ireland. LanguaL™ 2017

[5] Michael Gummelt and David Brody. Converting In-NOut Orders into a Structured Form. en. Mar. 2009.

[6] Rahul Agarwal and Kevin Miller. Information Extraction from Recipes. Tech. rep. 2011.

[7] Erica Greene. Extracting Structured Data From Recipes Using Conditional Random Fields. en-US. Apr. 2015