

Mobile Computing and Artificial Intelligence for Diet Management

Alessandro Mazzei¹(✉), Luca Anselma¹, Franco De Michieli²,
Andrea Bolioli³, Matteo Casu³, Jelle Gerbrandy⁴, and Ivan Lunardi⁵

¹ Dip. Informatica, Università di Torino, Torino, Italy
{alessandro.mazzei, luca.anselma}@unito.it

² Dip. Scienze Mediche, Università di Torino, Torino, Italy
franco.demichieli@unito.it

³ CELI s.r.l., Torino, Italy

{abolioli, mcasu}@celi.com

⁴ Gerbrandy s.r.l., Torino, Italy

jelle@gerbrandy.com

⁵ Synesthesia s.r.l., Torino, Italy

lunardi@synesthesia.com

Abstract. This paper proposes a software architecture for automatic diet management and recipes analysis. We devise a *virtual dietitian* that is able: (1) to recover the nutritional information directly from a specific recipe, (2) to reason over recipes and diets with *flexibility*, i.e. by allowing some forms of diet disobedience, and (3) to persuade the user to minimize these acts of disobedience.

1 Introduction

The increasing pervasiveness of technology is drawing some new scenarios on man-machine interaction. We are surrounded by computers and this revolutionary state of affairs can be exploited in order to enlarge the power of our senses [9]. Cloud computing technologies allow to consider mobile devices as a number of sophisticated sensors that enhance the human senses and, sometimes, change them in new surprising forms. The term “Quantified Self” has recently been adopted to indicate technologies that allow for a ubiquitous monitoring of human activities. Moreover, in recent years there has been a growing interest in using multimedia applications on mobile devices as *persuasive* technologies. For instance “UbiFit Garden” is a mobile application developed to encourage people to maintain a good level of physical activity in daily life. The users can see their weekly progress in the background of their smartphones [4].

The daily diet is one of the most important factors influencing diseases, in particular obesity. As highlighted by the World Health Organization, this factor is primarily due to the recent changes in the lifestyle [16]. The necessity to encourage world’s population toward a healthy diet has been sponsored by the FAO [13] and each nation has tried to specialize these guidelines by adopting strategies related to its *food history*; for instance, for USA refer to

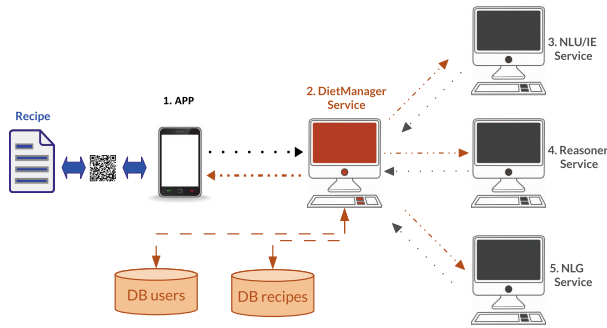


Fig. 1. The architecture of the diet management system.

<http://www.choosemyplate.gov/> and for UK <http://www.nutrition.org.uk>. The Italian Society for Human Nutrition has recently produced a prototypical study with recommendations for the use of specialized operators [1]. A less specialized Italian text, with recommendations easier to read, has been released in 2003 (http://nut.entecra.it/648/linee_guida.html). This complex scenario suggests the possibility to integrate the directives on nutrition in the daily diet of people by using multimedia facilities of mobile devices. A smartphone can be considered as an innovative super-sense that creates new kinds of interaction with the food. For example, we can imagine to print the QR code of a specific dish on a restaurant menu to allow to retrieve “on the cloud” that specific recipe. The information about the dish could be analyzed by an artificial dietitian in order to assess the compatibility of the course with a diet.

We want to design a *virtual dietitian*, i.e. an artificial intelligence (henceforth AI) software that is able to reason over recipes and diet with *flexibility*. In our opinion, a major strength of our design is the possibility for the user to not follow the recommendation produced by the system. So, the AI should make the system *tolerant* to some forms of diet disobedience, but it should also be *persuasive* to minimize these acts of disobedience. In this paper we describe a cloud-based software architecture developed in the project MADiMAN (Multi-media Application for DIet MANAGEMENT – <http://di.unito.it/madiman>) that uses Natural Language Processing (NLP) and Automatic Reasoning techniques for diet management. The paper is organized as follows. In Section 2 we describe the general architecture of the system. In Section 3 we describe three AI software modules designed for analyzing the recipes, for reasoning over the diet and for generating a message to motivate the user toward the right choice. Finally, Section 4 concludes the paper.

2 A Cloud Architecture for Diet Management

In our scenario the interaction between man and food is mediated by an intelligent recommendation system that, on the basis of various variables, encourages

or discourages the user to eat a dish that she intends to eat. The main factors that the system needs to account for are: (1) the diet that the user intends to follow, (2) the food that has been eaten in the last days, and (3) the nutritional values of the ingredients of the dish and its specific recipe.

The system is composed of five modules/web services (see Figure 1): a smartphone application (APP), a central module that manages the information flow (DietManager), a Natural Language Understanding / Information Extraction (NLU/IE) module, a reasoning module (Reasoner), a natural language generator module (NLGenerator). Upon a registration where the user provides her anthropometric data, the information flow follows this pattern:

1. The user, by using the APP, recovers the QR-code of a specific recipe, or finds the recipe using full-text search.
2. The APP, using the DietManager service, retrieves the user diet together with the list of the food that the user has eaten in the last days. Moreover, the DietManager retrieves the specific recipe of the specific dish.
3. The NLU/IE module computes the salient nutrition information about the specific course.
4. The Reasoner, using the user diet and the list of the food that has been eaten in the last days, produces the final recommendation about the dish for the user.
5. The NLGenerator uses the recommendation given by the Reasoner, produces an explanation for the user in plain natural language and/or in a multimedial format, e.g. by using icons.
6. The DietManager sends the result produced by the NLGenerator to the APP: the user will see this final result on her smartphone. If the user decides to eat the dish, the APP will send this information to the DietManager that will update the list of food eaten.

The DietManager plays the role of a hub for the stateless communications with the web services. So, all web services expose REST (REpresentational State Transfer) interfaces following the best practices for creating simple scalable web services. In the next Section we discuss the main issues of the three web services.

3 AI for Diet Management

The AI allows us to add flexibility to the web services. In Section 3.1 we describe the module for automatic analysis of recipes, in Section 3.2 we describe the reasoning system based on STP (Simple Temporal Problems), and in Section 3.3 we describe a persuasive NLG module.

3.1 NLP for Recipe Analysis

Many smartphone applications use food databases to store the nutritional content of a recipe, but such databases provide only a generic version of the recipes. So, these applications do not allow for a precise computation of the nutritional

values of a specific recipe. It is common knowledge that a recipe may vary from cook to cook, both in the quantity of the ingredients and in the methods of preparation. The automatic analysis of recipes is performed to accurately calculate the nutritional values of a specific recipe.

The NLU/IE module is based on an NLP pipeline with the following components: tokenizer, normalizer, sentence splitter, Part-of-Speech tagger, lemmatizer, Named Entities extractor, chunker. The pipeline implements the Apache UIMA (Apache Unstructured Information Management Architecture), and the architecture components are implemented as UIMA annotators [7]. Each component analyzes the text and produces a sort of annotation that will be used by the next component. The UIMA facilitates a rapid component integration, testing and evaluation. The goal of the NLU/IE module is the extraction of a set of pairs *ingredient*-quantity for each recipe. Moreover, the analyzer needs to recover some special information for the diet, such as the presence of particular cooking techniques (e.g. frying). The extraction of such information is rule-based, and occurs at end of the pipeline execution. Thus, the NLU/IE spots expressions of quantities in the text of the recipes, obtaining nutrients for the current recipe.

Although machine learning has been successfully used for similar tasks ([10]) we chose a rule-based approach for different reasons, namely a robust experience with rule-based systems by the team, and the fact of having at disposal a rich gazetteer for ingredients. Besides, a rule-based approach permitted to quickly set up the first experiments with results easily verifiable by hand, so that a linguist developer could analyze the major issues that emerged in order to report them.

The extraction can be performed by using a rule engine with condition-action rules acting on the analyzed text. For our experiments we used Drools (<http://www.drools.org>), a state-of-the-art Business Rules Engine developed by the JBoss Community. We integrated Drools engine into our implementation of the Apache UIMA framework. Besides being efficient, Drools is well maintained, comes under a Apache 2.0 License and is well integrated with common Java developers' tools such as Eclipse.

An example of pattern captured by the system has the form:

```
<number> ("g"|"gr"|"gram") "of" <ingredient>
```

(a number followed by an expression for grams, followed by an ingredient) and sets a new “Quantity” annotation on the document, positioned where the pattern was spotted.

We mention here two challenging issues we have encountered: 1. in order to normalize all quantities in grams, we have to know the specific weight for each liquid mentioned in text (water, wine, and so on); 2. some quantities occur often under vague expressions (“a glass of water”, “a handful of almonds”, etc.); this problem can be softened assuming standard quantities for the various expressions (e.g. a glass of water is of approximately 250 milliliters and weighs approximately 250 grams) – furthermore, in some cultures (e.g. anglo-saxon) such relations between expressions and volumes are quite standardized.

3.2 Reasoning over Recipes and Diet

Since our approach to automatic reasoning for diet management is based on the STP framework [5], first we introduce STP, then we describe how we exploit STP to reason on a diet and how we interpret the results from STP.

An STP [5] is composed by a conjunction of STP constraints of the form $c \leq x - y \leq d$, where x and y are temporal points and c and d are numbers (their domain can be either discrete or real). An STP constraint can be interpreted as “the temporal distance between the time points x and y is between a lower bound c and an upper bound d ”. It is also possible to impose strict inequalities and $-\infty$ and $+\infty$ can be used to denote the fact that there is no lower or upper bound, respectively. STP can be represented as a graph whose nodes correspond to the temporal points of the STP and whose arcs are labeled with the temporal distance between the points. The problem of determining the consistency of an STP can be solved efficiently by an all-pairs shortest paths algorithm such as Floyd-Warshall’s one, which also obtains the minimal network, that is the minimum and maximum distance between each pair of points.

Dietary recommendations are often expressed as Dietary Reference Values (henceforth DRVs) for (macro)nutrients that are recommended to be assumed for significant amounts of time. In particular, from the basal metabolic rate and physical activity level it is possible to estimate the total energy requirement and then obtain the DRVs for an individual. For example, if we assume that a patient has a total energy requirement of 2450 kcal/day, the DRVs in [1] provide that the appropriate amount of the macronutrients is of 260 kcal/day of proteins, 735 kcal/day of lipids and 1455 kcal/day of carbohydrates. In this section we focus on the total energy requirement; the macronutrients can be dealt with separately in the same way. The basic idea is to represent DRVs as STPs. More precisely, we use an STP constraint to represent both the DRVs and the actual values of the ingested food. For example, a recommendation to eat at lunch between 500 kcal and 600 kcal is represented by the STP constraint $500 \leq lunch_E - lunch_S \leq 600$, where $lunch_E$ and $lunch_S$ represent the end and the start of the lunch, respectively. To allow a user to make small deviations with regard to the “ideal” diet, we impose less strict constraints over the shortest periods (i.e., days or meals) and stricter constraints over the longest periods (i.e., months, weeks). For example the recommended energy requirement results in a constraint such as $2450 \cdot 7 \leq week_E - week_S \leq 2450 \cdot 7$ over a week and as $2450 - 10\% \leq Sunday_E - Sunday_S \leq 2450 + 10\%$ for a single day (e.g., Sunday). For single meals we further relax the constraints.

In order to effectively support the user, the system must also take into account what she actually eats. Therefore, a new, provisional, STP is generated where – in addition to the constraints derived from the DRVs – also the new STP constraints deriving from the meals that the user has actually consumed are added. The new constraints possibly restrict the values that the constraints in the original STP allow. The constraint propagation determines whether the new constraints are consistent and provides the new minimal network with the new implied relations. For example, let us suppose that the user on Sunday, Monday and

Tuesday had an actual intake of 2690 kcal for each day. This corresponds to adding to the STP the new constraints $2690 \leq \textit{Sunday}_E - \textit{Sundays}_S \leq 2690$, \dots , $2690 \leq \textit{Tuesday}_E - \textit{Tuesdays}_S \leq 2690$. Then, propagating the constraints of the new STP, we discover that (i) the intake is compatible with the diet (in fact the STP is consistent) and (ii) on each remaining day of the week the user has to assume between 2205 kcal and 2465 kcal.

We wish to provide the user with user-friendly information not limited to a visualization of the minimal network. The minimal network allows us to classify the macronutrients of the dish that the user intends to eat in the following cases: I.1) *permanently inconsistent*, I.2) *occasionally inconsistent*, C.1) *consistent and not balanced*, C.2) *consistent and well-balanced* and C.3) *consistent and perfectly balanced*. In the cases I.1 and i.2 the dish is inconsistent. In case I.1) the value of a macronutrient is inconsistent with the DRVs as represented in the user's diet. The dish cannot be accepted even independently of the other food she may possibly eat. In case I.2) the dish, taken in isolation, does not violate the DRVs, but, considering the past meals she has already eaten, it would preclude to be consistent with the diet. In cases C.1), C.2) and C.3) the value of the macronutrient is consistent with the diet, but a consistent but not balanced choice of a dish could have consequences on the rest of the user's diet because the user will have to "recover" from it. We assume that the mean value in the relative STP constraint is the "ideal" value according to the DRVs and we consider two parametric user-adjustable thresholds relative to the mean: according to the deviation with respect to the mean we classify the macronutrient as not balanced (case C.1), well balanced (case C.2) or perfectly balanced (case C.3). In particular, in the cases C.1) and C.2) we distinguish between lack or excess of a specific macronutrient for a dish: if it is lacking (in excess) with regard to the ideal value, we tag the dish with the keyword *IPO* (*IPER*). This information is exploited in the generation of the messages.

3.3 Motivating with NLG

In order to motivate an user to follow a good choice for eating, we reformulate the result of the reasoning into a textual form. This text will be combined with a suggestion in order to create a final output text. We designed a template-based system that, starting from the numerical results of the reasoning, returns a text message by following a number of textual patterns. That is, for each case in the reasoner output, we produce a textual template that will be filled with the specific content of the reasoning. These templates have been designed by basically considering three models of persuasion: (i) CAPTology, (ii) the Cialdini's Six Principles of Influence, (iii) the persuasive use of the emoticons [3, 6, 8].

In CAPTology (Computers As Persuasive Technologies), the principal idea is that the computer is perceived in three coexisting forms, Tool-Media-SocialActor. As a tool, the computer can enhance the capabilities of a user: our system calculates the nutritional contents of a dish, and so it enhances the ability to correctly judge the compatibility of the dish with a diet. As a media the

Table 1. The persuasive message templates.

C	D	Message Template	Translation
I.1	IPO	Questo piatto non va affatto bene, contiene davvero pochissime proteine!	This dish is not good at all, it's too poor in proteins!
I.2	IPO	Ora non puoi mangiare questo piatto perché è poco proteico. Ma se domenica mangi un bel piatto di fagioli allora lunedì potrai mangiarlo.	You cannot have this dish now because it doesn't provide enough proteins, but if you eat a nice dish of beans on Sunday, you can have it on Monday.
C.1	IPO	Va bene mangiare le patatine ma nei prossimi giorni dovrai mangiare più proteine.	It's OK to eat chips but in the next days you'll have to eat more proteins.
C.2	IPO	Questo piatto va bene, è solo un po' scarso di proteine. Nei prossimi giorni anche fagioli però! :)	This dish is OK, but it's a bit poor in proteins. In the next days you'll need beans too! :)
C.3	-	Ottima scelta! Questo piatto è perfetto per la tua dieta :)	Great choice! This dish is perfect for your diet :)

computer “provides experience”: in our system, the human memory is enhanced by the reasoner, which indirectly reminds the user what she ate in the last days. As a social actor the computer creates an empathic relationship with the user reminding her the “social rules”. Specifically, we designed the templates to guide the user towards the choice of a balanced meal, convincing her, by using a *friendly* recommendations, to follow the diet that herself decided. The six principles of Cialdini are: (1) Reciprocity: “people feel obligated to return a favor”, (2) Scarcity: “people will value scarce products”, (3) Authority: “people value the opinion of experts”, (4) Consistency: “people do as they said they would”, (5) Consensus: “people do as other people do”, (6) Liking: “we say yes to people we like”. In particular, our templates follow two of these principles, i.e. authority and consistency. Finally, in some messages we decided to use emoticons, since they can increase the communicative strength of a message by creating a tone of friendship, in particular for positive messages [6].

We use five templates to communicate the five cases in the reasoner output: in Table 1 we report the cases obtained by the reasoner output interpretation (column C), the direction of the deviation (column D), the Italian templates, and their rough English translation. For space reason, we do not describe the algorithm used in the generation module to combine the three distinct outputs of the reasoner on the three distinct macronutrients (i.e. proteins, lipids and carbohydrates). In short, the messages corresponding to each macronutrient need to be *aggregated* into a single message. A number of constraints related to coordination and relative clauses needed to be accounted for [14].

4 Conclusions

In this paper we proposed a software architecture for diet management: we discussed the main issues related to the modules used to analyze a recipe, to reason over the diet and to generate a message for the user.

There is a number of academic studies related to our project, e.g. [2, 11, 12, 15], and there is also a great number of smartphone applications related to nutrition on the Italian and international markets, e.g. *DailyBurn*, *Lose It!*, *MyNetDiary*,

Weight Watchers. However, our dietary system presents three elements of novelty: (1) the recovery of nutritional information directly from the recipes by using NLU/IE techniques, (2) the use of automatic reasoning as a tool for verifying the compatibility of a specific recipe with a specific diet, and (3) the use of NLG techniques to produce the answer.

In the next future we intend to test our system by using a simulation. We plan to use a real database of recipes and a number of realistic diets into the context of a hospital.

References

1. LARN - Livelli di Assunzione di Riferimento di Nutrienti ed energia per la popolazione italiana - IV Revisione. SICS (2014)
2. Balintfy, J.L.: Menu planning by computer. *Commun. ACM* **7**(4), 255–259 (1964)
3. Cialdini, R.B.: *Influence: science and practice*. Pearson Education, Boston (2009)
4. Consolvo, S., et al.: Activity sensing in the wild: a field trial of ubifit garden. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2008*, pp. 1797–1806. ACM, New York (2008)
5. Dechter, R., Meiri, I., Pearl, J.: Temporal constraint networks. *Artif. Intell.* **49**(1–3), 61–95 (1991)
6. Derks, D., Bos, A.E.R., von Grumbkow, J.: Emoticons in computer-mediated communication: Social motives and social context. *Cyberpsy., Behavior, and Soc. Networking* **11**(1), 99–101 (2008)
7. Ferrucci, D., Lally, A.: UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.* **10**(3–4), 327–348 (2004). <http://dx.doi.org/10.1017/S1351324904003523>
8. Fogg, B.: *Persuasive Technology. Using computers to change what we think and do*. Morgan Kaufmann Publishers / Elsevier, San Francisco (2002)
9. Furht, B. (ed.): *Handbook of Augmented Reality*. Springer (2011)
10. Greene, E.: Extracting structured data from recipes using conditional random fields (2015). <http://open.blogs.nytimes.com/2015/04/09/extracting-structured-data-from-recipes-using-conditional-random-fields>
11. Iizuka, K., Okawada, T., Matsuyama, K., Kurihashi, S., Iizuka, Y.: Food menu selection support system: considering constraint conditions for safe dietary life. In: *Proceedings of the ACM Multimedia 2012 Workshop on Multimedia for Cooking and Eating Activities, CEA 2012*, pp. 53–58. ACM, New York (2012)
12. Mankoff, J., Hsieh, G., Hung, H.C., Nitao, E.: Using low-cost sensing to support nutritional awareness. In: Borriello, G., Holmquist, L.E. (eds.) *UbiComp 2002*. LNCS, vol. 2498, pp. 371–376. Springer, Heidelberg (2002)
13. Nishida, C., Uauy, R., Kumanyika, S., Shetty, P.: The joint WHO/FAO expert consultation on diet, nutrition and the prevention of chronic diseases: process, product and policy implications. *Public Health Nutrition* **7**, 245–250 (2004)
14. Reiter, E., Dale, R.: *Building Natural Language Generation Systems*. Cambridge University Press, New York (2000)
15. Siek, K., Connelly, K., Rogers, Y., Rohwer, P., Lambert, D., Welch, J.: When do we eat? an evaluation of food items input into an electronic food monitoring application. In: *Pervasive Health Conference and Workshops*, pp. 1–10 (2006)
16. World Health Organization: Global strategy on diet, physical activity and health (WHA57.17). In: *75th World Health Assembly* (2004)