Presented at the DLSU Research Congress 2015
De La Salle University, Manila, Philippines
March 2-4, 2015

Towards Automatically Building a Knowledge Base of Dietary Needs to Address Medical Conditions

Dominic William Elayda III, Justin Ervin Garcia, Danielle Lladoc,
Paolo Uy, and Nathalie Rose Lim-Cheng
De La Salle University-Manila
2401 Taft Avenue
Manila, Philippines
** Nathalie Rose Lim-Cheng: nats.lim@delasalle.ph*

Abstract: For years, health articles and dietary advice have been provided by dieticians and clinical nutritionists on different online media. Also, many websites provide information and recommendations on how to manage their weight. Lately, there are software applications that generate recommendations of what food to consume for a healthier life. However, these do not consider the medical conditions or dietary needs of the user. That is, for example, if the user is anemic, a recommendation system should recommend foods rich in iron. Also, ideally, if the user is diabetic, a recommendation system would not recommend sugar-rich fruits, like mango. However, for a recommendation system to be able to generate a suitable suggestion that would address medical conditions, a knowledge base of dietary needs and constraints has to be built. This can be built manually, but this is time consuming if experts are to be asked to encode all the information. Therefore, this research is an attempt to alleviate the tedious task of encoding via automatic extraction of information from medical texts (e.g., from MedLine Plus) and web articles (e.g., from WebMD). This extraction process produces candidate entries that will be saved into the knowledge base. But prior to storing, a validation process allows an expert to verify the correctness.

This paper presents the components and the processes involved in the extraction to associate the necessary nutrients for medical conditions to food items that contain them. Initial testing using health-related articles available on the internet show promising results.

Key Words: Ontology population; information extraction; medical conditions; nutritional needs

1. INTRODUCTION

Dieticians and clinical nutritionists have published dietary advice in websites, blogs, online publications and peer-reviewed journals. These readily available online information presents a number of issues. A large portion of the results returned by a typical web search query is usually irrelevant (Lieto, 2008). Thus, when performing a web search, one must go through the steps of determining relevant keywords and verifying the validity and authenticity of the document. Health informatics, a discipline that uses computers and Information and Communication Technology (ICT) in order to improve the usage and management of information in health and biomedicine (UKCHIP,n.d.), aims to solve this problem. Most of

the information available comes in the form of unstructured texts. In order for this information to be used, it needs to be processed and represented in a machine-readable form (Faria & Girardi, 2011). For knowledge-based systems, ontologies are created in order to formally represent a body of knowledge in a particular domain as objects and their relationships with each other (Gruber, 1993). The construction of this ontology is done through a process called ontology population.

Our research involves designing and implementing a system that can automatically generate candidate entries to populate an ontology that associates the necessary nutrients from different medical conditions to food items that contain them by means of information extraction (IE). Experts can view the candidate entries prior to storing into the ontology.

Section 2 lists a brief overview of the different ontology population systems that currently exist today. In Section 3, the design for the ontology that will be used as the proposed knowledge based is presented in detail. Section 4 discusses the architecture of the system and the different components it is made up of. In Section 5, we present some initial testing results. Finally, section 6 presents areas of improvements and possible areas for further research.

2. EXISTING SYSTEMS

There are already a number of existing systems designed to populate an ontology for different domains. These systems use varied techniques and methodologies in order to populate these ontologies. Ontology population can either be manual, semi-automatic, and automatic. Tools for semi-automatic, and automatic ontology population are developed to avoid the cost that manual approach needs. In this section, three existing ontology population systems are discussed – SPRAT, OntoSophie and the Philippine Medicinal Plant Ontology Population System.

SPRAT is a tool for automatic semantic pattern-based ontology population (Maynard et. al., 2009). It identifies patterns from the input text which makes it possible for the system to extract a variety of entity types and relations between them, and also re-engineer them into concepts and instances in an ontology. This is made possible by combining different aspects from traditional named entity recognition, ontology-based information extraction and relation extraction. SPRAT can either create an ontology from scratch or modify an existing one. The system was developed in GATE, an architecture for language engineering. It makes use of a GATE plugin called NEBOnE (Named Entity Based Ontology Editing) in its ontology editing feature. NEBOnE is used in processing natural language text and manipulating ontologies.

OntoSophie is a system for semi-automatic population of ontologies with instances from unstructured text (Celjuska & Vargas, 2004). It learns extraction rules from manually annotated text, also known as supervised learning, and apply those rules to the input text to populate an ontology. The system goes through three phases when populating an ontology namely, annotation, learning, and extraction and ontology population. The system has three main components namely Marmot, Crystal, and Badger which performs the phases stated earlier, respectively. The annotation phase is where a set of plain text or HTML documents are annotated with XML tags and assigned to one of the predefined classes in the ontology. The learning phase is where the input is parsed and extraction rules are generated. The last phase, extraction and ontology population, is where entities are extracted from the input text and instances are constructed into the ontology.

The Philippine Medicinal Plant Ontology Population System is a system for semi-automatic population of the ontology of Philippine Medicinal Plants from online text (Lim-Cheng et. al., 2014). Data is extracted from various medicinal plant articles by undergoing a set of back-end and front-end components. It starts by retrieving relevant articles using Crawler4J web crawler. The outputted text file is passed to the ANNIE English Tokenizer to identify the article contents' token types. The system tags tokens using a tagger model trained using the GENIA Corpus and determines its corresponding Penn Tree Bank Part-of-Speech using the LingPipe Tokenizer. The system uses annotators, ANNIE Gazetteer and JAPE Rules, to annotate entities. The sentence patterns serve as a basis for the construction of rules. To resolve referencing problems in sentences, the system uses Anaphora resolution. The processed data are stored in a template which is presented to the user for validation. After validating the contents of the template and determining which information are to be added onto the ontology, the user has the option to save, add/edit and search the ontology.

3. ONTOLOGY DESIGN

The ontology design used for this system is shown in Figure 1. The goal of this ontology is to map medical conditions related to nutrition to their

symptoms, and to show their relationship to the different nutrient constituents of food items. It is implemented in OWL, and is built from scratch, although it borrows some designs from existing ontologies on food, nutrition and medicine.

Medical conditions are retrieved from Disease and SYMP (symptoms) ontologies. The food and their nutritional contents are retrieved from USDA National Nutrient Database. Since information comes from different sources, the relationship among all of these (conditions with the food or the actual nutrient or conditions with restrictions) are determined from processing articles found on medical websites.

3.1 Ontology Concepts

The ontology consists of four (4) main concepts, namely Food, Condition, Nutrient and Symptom.

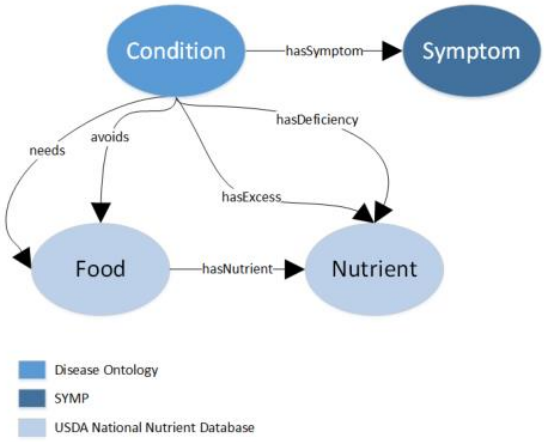


Fig. 1. Ontology Design

3.1.1. Food

Food represents the food items contained in the ontology. It has the property *hasNutrient* which specifies a specific type of Nutrient that it contains.

3.1.2 Condition

Condition represents the medical conditions in the ontology. It has the following properties:

- *hasSymptom* specifies whether the Condition is associated with a specific type of Symptom.
- *needs* specifies whether the Condition is needs a particular Food item. This relationship is established when the Food item contains a Nutrient that the Condition is deficient in.
- *avoids* specifies whether the Condition should avoid a particular Food item. This relationship is established when the Food item contains a Nutrient that the Condition is excessive in or is sensitive to.
- *hasDeficiency* specifies whether the Condition is deficient in a particular type of Nutrient.
- *hasExcess* specifies whether the Condition is excessive or sensitive to a particular type of Nutrient.

3.1.3 Nutrient

Nutrient represents the nutrients that the Food instances contain.

Fields

- *name* contains the name of the nutrient according to the USDA National Nutrient Database
- *synonym* contains the other names that this nutrient may appear as or be known by.
- *value* contains a number representing how much of the nutrient is contained in 100g of the Food instance it is associated with, based on the values in the USDA National Nutrient database.
- *minAmount* refers to the minimum intake amount of the nutrient for normal, healthy adults as specified by nutritionists
- *recommendedAmount* refers to the recommended intake amount of the nutrient for normal, healthy adults
- *maxAmount* contains the maximum intake amount of the nutrient for normal, healthy adults

3.1.4 Symptom

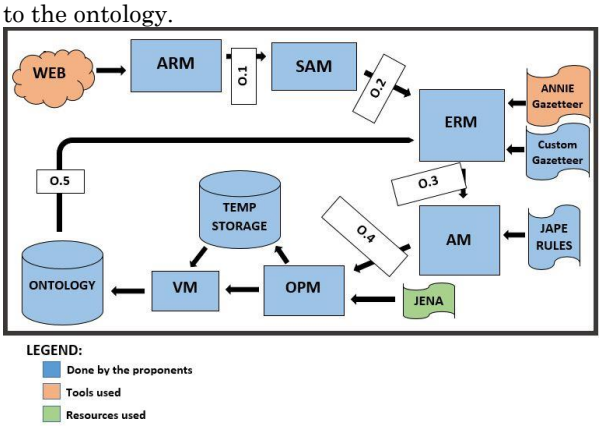
Symptom represents the symptoms of a particular Condition. It is possible that one Symptom can be associated with different Conditions.

4. OUR SYSTEM

Figure 2 shows the system architecture designed to extract candidate entries to populate the ontology. The information needed by the system to populate the ontology were retrieved from health and nutrition websites.

The system operates in three stages. The first stage is the information retrieval stage, which is done by the Article Retrieval Module (ARM). Next is the preprocessing stage which consists of three different modules: the Syntactic Analysis Module (SAM), Entity Recognition Module (ERM) and the Annotation Module (AM). Once the data has been preprocessed, the ontology is ready to be created and populated. This is where the 3rd stage begins and includes the Ontology Population Module (OPM) and the Validation Module (VM). This ontology population process is an automated process. But to ensure reliability of the data, experts in the domain

verify the extracted instances before they are added to the ontology.



Label Legend:
ARM: Article Retrieval Module
SAM: Syntactic Analysis Module
ERM: Entity Recognition Module
AM: Annotation Module
OPM: Ontology Population Module
VM: Validation Module
O.1. Web Articles in text files
O.2. Text with PoS tags in XML
O.3. Text with entity type tags in XML
O.4. Text with relationship tags in XML
O.5. Food list

Fig. 2. System Architecture

4.1 Article Retrieval Module

This module is a web crawler that retrieves articles from a list of predefined websites (e.g., Web MD, MedLine Plus, etc.). Once the crawler has started, it retrieves the contents of the web pages one by one, and saving them as text files for preprocessing. Once the web pages are retrieved, this module also performs minimal cleaning on the web pages, removing unnecessary markup tags. Figure 3 shows a sample excerpt of a retrieved article after it has gone through the cleaning process.

```
Scurvy is a disease caused by a lack of vitamin C in the diet.
…
Scurvy is treated with daily vitamin C (ascorbic acid)
tablets.
Symptoms usually improve quickly.
```

Fig. 3. Sample article from WebMD after cleaning

4.2 Syntactic Analysis Module

The syntactic analysis module performs the initial preprocessing on the raw textual documents collected by the article retrieval module from the web through tokenization and part-of-speech (PoS) tagging. This involves splitting up the document into parts called "tokens" and labeling it with the proper PoS. For this module, the LingPipe Tagger is used. It has models from GENIA and MedPost corpora which are English biomedical documents making the tagger ideal for processing medical documents. This module's output contains the input text with the words or phrases tagged with their corresponding lexical category in XML format. Figure 4 shows a sample output of this module from processing the article in Figure 3.

```
<ARTICLE>
<NN>Scurvy</NN> <VBZ>is</VBZ> <DT>a</DT>
<NN>disease</NN> <VBD>caused</VBD> <IN>by</IN>
<DT>a</DT> <NN>lack</NN> <IN>of</IN>
<NN>vitamin</NN> <NN>C</NN> <IN>in</IN>
<DT>the</DT> <NN>diet</NN> <SPECIAL>.</SPECIAL>
…
<NNP>Scurvy</NNP> <VBZ>is</VBZ>
<VBN>treated</VBN> <IN>with</IN> <JJ>daily</JJ>
<NN>vitamin</NN> <NN>C</NN>
<SPECIAL>(</SPECIAL> <JJ>ascorbic</JJ>
<NN>acid</NN> <SPECIAL>)</SPECIAL>
<NNS>tablets</NNS> <SPECIAL>.</SPECIAL>
<NNS>Symptoms</NNS> <RB>usually</RB>
<VB>improve</VB> <RB>quickly</RB>
<SPECIAL>.</SPECIAL>
</ARTICLE>
```

Fig. 4. Scurvy article that has been tagged by the Syntactic Analysis Module

4.3 Entity Recognition Module

In this module, elements of the input text are classified and annotated with their corresponding entity types. Here each token is analyzed and tagged with label from a set of predefined entity types. This determines how and where in the ontology these entities should be stored. This module accepts as input an XML file that is the previous module's output and outputs a modified XML file with named entities respectively annotated. A tool called ANNIE is used with a custom gazetteer to perform the entity recognition. An excerpt of this module's output from processing the document in Figure 4 can be seen in Figure 5.

```
<ARTICLE>
<Entity Type="Condition">Scurvy</Entity>
<VBZ>is</VBZ> <DT>a</DT> <NN>disease</NN>
<VBD>caused</VBD> <IN>by</IN> <DT>a</DT>
<NN>lack</NN> <IN>of</IN> <Entity
Type="Nutrient">vitamin C</Entity> <IN>in</IN>
<DT>the</DT> <NN>diet</NN> <SPECIAL>.</SPECIAL>
…
<Entity Type="Condition">Scurvy</Entity>
<VBZ>is</VBZ> <VBN>treated</VBN> <IN>with</IN>
<JJ>daily</JJ> <Entity Type="Nutrient">vitamin
C</Entity> <SPECIAL>(</SPECIAL> <Entity
Type="Nutrient">ascorbic acid</Entity>
<SPECIAL>)</SPECIAL> <NNS>tablets</NNS>
<SPECIAL>.</SPECIAL> <NNS>Symptoms</NNS>
<RB>usually</RB> <VB>improve</VB> <RB>quickly</RB>
<SPECIAL>.</SPECIAL>
```

```
</ARTICLE>
```

Fig. 5. Excerpt from the output of Entity Recognition Module

4.4 Annotation Module

This module is where grammar rules or lexico-syntactic patterns from premade patterns are applied. The files containing the grammar rules are fed to the entity recognition module along with the XML file from the previous module to perform the annotation. These rules specify the entity types to be identified when certain grammar patterns are encountered in the input. The output of this module is a modified XML file with entity relation annotations and additional entity type annotations. These annotations specify how the entities will be linked together in the ontology. The ANNIE tool is also used in interpreting the JAPE rules and making the changes to the XML file as needed. Figure 6 shows the output of this module, using the sample in Figure 5 as input.

```
<ARTICLE>
<condition>Scurvy</condition>
<synonym></synonym>
<symptom>
        <value>weakness</value>
        <value>swollen joints</value>
...
<needNutrient>
        <value>vitamins c</value>
</needNutrient>
<avoidNutrient></avoidNutrient>
</ARTICLE>
```

Fig. 6. Excerpt from the output of the Annotation Module

4.5 Ontology Population Module

This module creates instances of common medical conditions affecting adults and their details, like nutrient deficiency or nutrient excess of the condition, its symptoms, and recommended food items and food items to avoid. Depending on the output of the next module (discussed in Section 0), this module might create new instances to add to the ontology or disregard some parts of the output. The module accepts as an input an XML file from the previous module, stores this XML file in a temporary storage by making an copy of this XML file in a specified directory, and communicate these instances to the next module, the validation module.

4.6 Validation Module

This module accesses the instances created by the ontology population module and presents it to the user for validation. The user is able to browse the list of instances, add, reject and discard instances, or skip instances for future validation.

5. INITIAL TESTING

For initial testing of the system, five (5) articles crawled from MedLine and another five (5) articles crawled from Biovision were manually annotated according to the four (4) different concepts discussed in the ontology design used (discussed in Section 3.1). These manually annotated documents serve as the gold standard for evaluating the results of this testing.

The same ten articles were processed by the rest of the modules. Precision, recall, and F-measure are the metrics used to measure the correctness of the annotations done by the system.

The system is capable of automatically retrieving articles from medical websites given a seed URL and is also able to preprocess the retrieved articles by annotating them with their respective entity concepts in the ontology.

The syntactic analysis module was able to successfully tokenize the articles, assign part-of-speech tags for each token, and properly place the tags and produce the XML file for each of these articles. Some of the tags used by the LingPipe tagger were modified or replaced in order for the succeeding modules to properly parse the output of this module. Tags that are special characters were replaced while tags that contain special characters were modified.

The entity recognition module was able to recognize the named-entities in the articles that are in the custom gazetteer but there are still some limitations. As of the moment the entity recognition module still needs to be improved to recognize instances of the gazetteer entries that are in the article even if it is not the same case (i.e., Scurvy, SCURVY, etc.). The module's ability to recognize named-entities is also limited by how big the gazetteer used is.

Currently, several JAPE rules have already been created to annotate entities. The rules were made to extract and annotate synonyms, symptoms, food needed, food to avoid, deficient nutrients, and excess nutrients for different medical conditions. The results were stored into an XML file.

It is these set of XML files that were compared with the gold standard. The results can be seen in Table 1 to 4, where:

TP = the number of "true positives" or the instances that were correctly extracted

FP = the number of "false positives" or the instances that were extracted but were incorrect

FN = the number of "false negatives" unsuccessfully extracted (i.e. "missing") information.

Table 1. List of results for MedlinePlus

Concept	Actual (gold)	Retri-eved	TP	FP	FN
Food	28	14	11	3	17
Condition	69	51	19	32	50
Nutrient	59	47	27	20	32
Symptom	59	3	1	3	55

Table 2. List of results for MedlinePlus (cont.)

Concept	Precision	Recall	F-measure
Food	0.786	0.393	0.524
Condition	0.372	0.275	0.317
Nutrient	0.574	0.458	0.509
Symptom	0.25	0.018	0.034

Table 3. List of results for Infonet-Biovision

Concept	Actual (gold)	Retri-eved	TP	FP	FN
Food	83	62	62	0	21
Condition	50	49	42	7	8
Nutrient	99	83	60	23	39
Symptom	44	4	0	4	44

Table 4. List of results for Infonet-Biovision (cont.)

Concept	Precision	Recall	F-measure
Food	1.000	0.747	0.855
Condition	0.857	0.840	0.848
Nutrient	0.723	0.606	0.659
Symptom	0	0	0

Based on the results for MedlinePlus, the system was not able to fully annotate Food and Symptom entities due to problems regarding the JAPE rules. The rules can identify symptoms but it annotates them per sentence or paragraph. Likewise, the results for Infonet-Biovision was also not able to annotate Symptom entities. The computed F-measure for MedlinePlus is lower than the F-measure for Infonet-Biovision. It can mean that the JAPE rules work better for MedlinePlus articles than for the other website. With this, there is a need to review the existing rules in order to complement both medical websites.

7. CONCLUSION

Though results in the initial testing look promising, there is a need to perform more tests to isolate the cause of discrepancies. For example, further testing of the syntactic analysis module should be done to verify the correctness of the part-of-speech tags that it assigns to the words in the articles.

In addition, more keywords should be included in the system to be able to increase the accuracy of annotating different entities. Also, further modification and improvements should be done for the JAPE rules to further increase the accuracy of annotating named entities. Once improved, more testing will be done. After which, extracted entries can be presented to the experts in the validation module. In this module, it should ensure that duplicate entries will be merged in the presentation for the expert validation.

On the other hand, further work can be done by exploring whether employing machine learning techniques could improve results of extraction and ontology population. Lastly, once the ontology is populated with sufficient instances, medical nutrition, and dietary recommendation or question-answering systems can be developed.

8. REFERENCES

Celjuska, D., & Vargas-Vera, M. (2004). OntoSophie: A Semi-Automatic System for Ontology Population from Text. *International Conference on Natural Language Processing (ICON) 2004*. Hyderabad, India.

Faria, C., & Girardi, R. (2011). An Information Extraction Process for Semi-automatic Ontology Population. *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*, 319-328. Berlin: Springer Berlin Heidelberg.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition, 5*(2), pp. 199-220.

Lieto, A. (2008). Manually vs. Semi-automatic Domain-specific Ontology Building. *Master's Thesis*. University of Salerno, Italy.

Lim-Cheng, N. R., Co, J. R., Gaudiel, C., Umadac, D., & Victor, N. (2014). Semi-Population of Ontology of Philippine Medicinal Plants from On-Line Text. *DLSU Research Congress 2014*. De La Salle University, Manila, Philippines.

Maynard, D., Funk, A., & Peters, W. (2009). SPRAT: A Tool for Automatic Semantic Pattern-Based Ontology Population. *International Conference for Digital Libraries and the Semantic Web 2009*. Trento, Italy.

UKCHIP. (n.d.). *What is Health Informatics?* Retrieved February 9, 2014, from UKCHIP Website: http://ukchip.org/?page_id=1512