I - Introduction

Believe that everybody has done such a thing, when we feel uncomfortable, we will go to Google to search for our symptoms to know which disease that we get possibly. But in general, we get the results who are some long paragraphs, we have to read it and then get the results. We put forward a solution to create a system who can deduce the disease that people get by entering the symptoms and the position of our body that we feel ailing.

II - Analyse of project

Our project will be divided into the following steps:

1 - Data acquisition

We find some sites that specialize in consulting the disease, we can get information from these sites through the crawler, the overall information will be divided into three categories, the body part, the symptoms and the diseases.

2 - Analysis of the input

Each person describes the symptoms in different ways, we need to analyze users' inputs (sentences or phrases) to get the keywords or the synonyms of input so that our system can find easier the users' input later to obtain the exact results.

3 - Construction of the ontology and rules

We build the ontology through the three types of data that we get, according to these three types of data, we build rules with SWRL.

4 - Results analysis

Due to the different input of each user, the symptoms entered by users are not necessarily in full compliance with a disease, maybe symptoms are consistent with a variety of diseases. In this case, after the semantic analysis, if user's input is fully correspond to our rule base, our platform will return the disease which correspond exactly user's symptoms, and we will also give other possibilities of diseases that user's symptoms correspond part of syptoms of those diseases. But if the user's input are incomplete, it can cause that symptoms can not fully satisfy our established rules, our system will return all the possible rows for the use, the order of possible diseases is according to the number of matches.

III - Data acquisition

At present, we use two ways to obtain the relevant data :

1 - Implementing crawlers to crawl information on the similar sites

https://www.mayoclinic.org/symptom-checker/select-symptom/itt-20009075

For example, the above website which uses a passive choice of users to chosse their own symptoms in order to obtain the same name of disease in their database. By this, we can effect the reptile simulation of user operations using crawl to access to all of its symptoms, as well as the combination of different symptoms and the introduction of the disease name.

2 - Using API to achieve data acquisition
apimedic is a well-known symptom and name of symptom online checker in the field of symptom. We can call the API of this platform to get the information we need.
For example, when we use the Get method to get the api: /symptoms

```
[
  {
    "ID": 10,
    "Name": "Abdominal pain"
  },
  {
    "ID": 238,
    "Name": "Anxiety"
  },
  .....
]
```

When we get the disease, we can get according to its ID which disease is caused by the disease: api
for ex: ID: 10, Abdominal pain and ID 238 will cause:

```
[
  {
    "ID": 54,
    "Name": "Reduced appetite"
  },
  {
    "ID": 52,
    "Name": "Sleeplessness"
  }
]
```

IV - User input analysis

When people enter their symptoms, maybe they will use a sentence, some words, different words have different deformations, every one’s input will be different, we need APIs to analyse input of users to get the keyword to search in our system.

1 - JCSEG
Jcseg is a lightweight word segmenter based on the mmseg algorithm which integrates functions like key extraction, key phrase extraction, critical sentence extraction and article auto digests. It provides a web server based on Jetty which permits different languages to call this API. Jcseg provides also the latest version of “lucene”, “solr”, “elasticsearch” word segmentation interface.

For example, the interface “lucene” provides the algorithm and the principle of english word segmentation, it can :

- Separate from the space / symbol / paragraph to get the phrase

With the regular expression, this step is easily realized.

- Extract stemming

Stemming is a Western language-specific treatment, such as the English word has a singular and plural deformation, -ing and -ed deformation, but for the system, these word should be the same word. Such as apple and apples, doing and done are seen as the same word, the purpose of extracting stem is to merge these metamorphosis.

2 - LexicalSynonym
LexicalSynonym API provides users with synonyms query service, it uses the whole network data mining to find massive synonyms and it does iterative updates to ensure that the synonymy effect is always advancing with the times.

JCSEG API can help our system find the keyword to do the search in our system, LexicalSynonym API can find the synonyms of users’ inputs to search the input after analyzing by APIs in our system.

V - Construction of Disease Diagnosis Rules Based on SWRL

1 - Ontology
There are three ways to establish ontology : “top-down”, “bottom-up” and “synthesis”.
“Top-down” approach begins with the largest concept, then we add its subclass and gradually refine the concept. “Bottom-up” approach begins with the smallest concept and organizes these small concepts into large concepts. The synthesis method is summarized and refined by introducing the obvious concepts.
We used the synthesis method to build conception of our ontology. Human disease can occur in different parts of the body, each disease has different symptoms. We use data that we get from the web to build our ontology grouping by human body parts, disease and symptoms .

We divided the data into these categories :

Body	External	Head, yes, legs, hands etc..
	Internal	Heart, kidney
Symptom	External	bleed, wound inflammation, wound pus, swollen
	Internal	headache, stomach pain, stomach acid, bloating, nausea etc..
Disease	Heart disease, cancer, gastric ulcer, cold, fever, etc...	

2 - ABOX
All of our data have the definition our their properties. We show you some examples :

- properties of symptoms

has_symptoms (x,a) : This person x has the symptom a.

- properties of parts of our body

has_body (x,b) : This person feels uncomfortable at the b(part of body).

- properties of disease

has_disease(x,c) : This person x has disease c.

- class of disease

cold(m) : m is disease cold

3 - TBOX
Example :
cold(x) ☰ has_symtom(Paul, NasalCongestion) +has_symptom (Paul, headache)
fever(x) ☰ cold(x) ^ has_symptom(Paul,highBodyTemperature)

4 - SWRL
In the field of semantic web, the ontology and the semantic reasoning and other related technologies provide us reasoning methods, we will use SWRL (Semantic Web Rule Language) and the OWL (Ontology Web Language) to achieve semantic level of reasoning.

SWRL (Semantic Web Rule Language) is a language to present rules by semantic way, SWRL combines the OWL and RuleML (Rule Markup Language), part of the SWRL rule concept comes from RuleML. When we create rules, by using the combination of the OML and RuleML, SWRL can use directly the conception and properties defined in the ontology, but if we don't use SWRL, we need some other rules to define the relation between these classes.

RuleML is also a regular modeling language, it is used to share and publish rules based on XML on the web. The word "head" means the result of the reasoning, and the word "body" represent the reasoning premise. For example, body : hasSon(A,B), this means that B is the son of A , hasBrother(B,C), this present that C is the brother of B.

When we use the SWRL to write the rules, every rule is composed by body and head (antecedent and consequent), every antecedent and consequent is composed of multiple elements. By using the SWRL, we can create a rule to describe the relation between A and C, head : hasUncle(A,C) : C is uncle of A.

Body

hasSon(A,B)

hasBrother(B,C)

Head

hasUncle(A,C)

4 - Reasoning rules to deduce the diagnosis based on SWRL

We use the data that we got from Internet and create rules by using SWRL, we take an example to show you how we create rules.

As we know, people who have diabetes eat more, drink plenty of water and they have more urine, but they lose weight, we use SWRL to create the rule for these symptoms and the disease diabete.

Body : (antecedent)

has_symptom(Paul, EatMore) : Paul eats so much.

has_symptom(Paul, DrinkMuchWater) : Paul drinks so much water.

has_symptom(Paul, MoreUrine) : Paul has more urine.

By using SWRL, we have rules :

Head : (consequence)

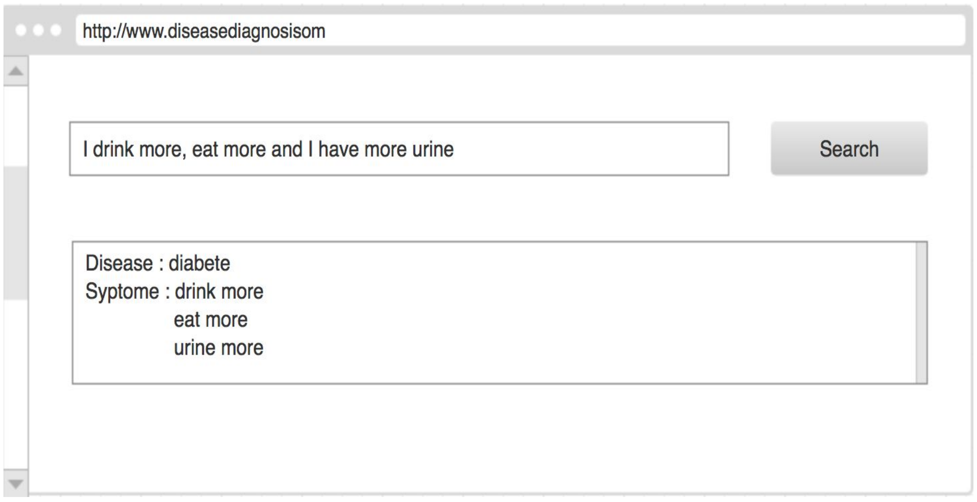
has_symptom(Paul, EatMore) ^ has_symptom(Paul, DrinkMuchWater) ^ has_symptom(Paul, MoreUrine) → has_disease (Paul,Diabete)

We use this way to create our rules base. There are many rules in our rule base by using data from Internet, when users search their symptoms, our machine can find automatically in our base to give user the answer. Because of different input of each user, maybe the symptoms entered by users can not fully conform our base, some symptoms can appear in several diseases. According to these reasons, If user's input is fully correspond to our rule base, our platform will return the disease which correspond exactly user's symptoms, and we will also give other possibilities of diseases that user's symptoms correspond part of syptoms of those diseases. If user's input can not conform to any rule in our base, we will list all possible diseases, it means, for all the rules, the part of conditions (symptoms) contains the symptoms entered by the user, we will show the corresponding disease.

We will use Jess (Java Expert System shell) for semantic reasoning, the API is small, processing quickly. Our platform is a website, we will use J2EE to build, it is compatible with HTML, PHP, js etc and with all the Java API, it is very convenient for us.

VI - Realisation of our platform

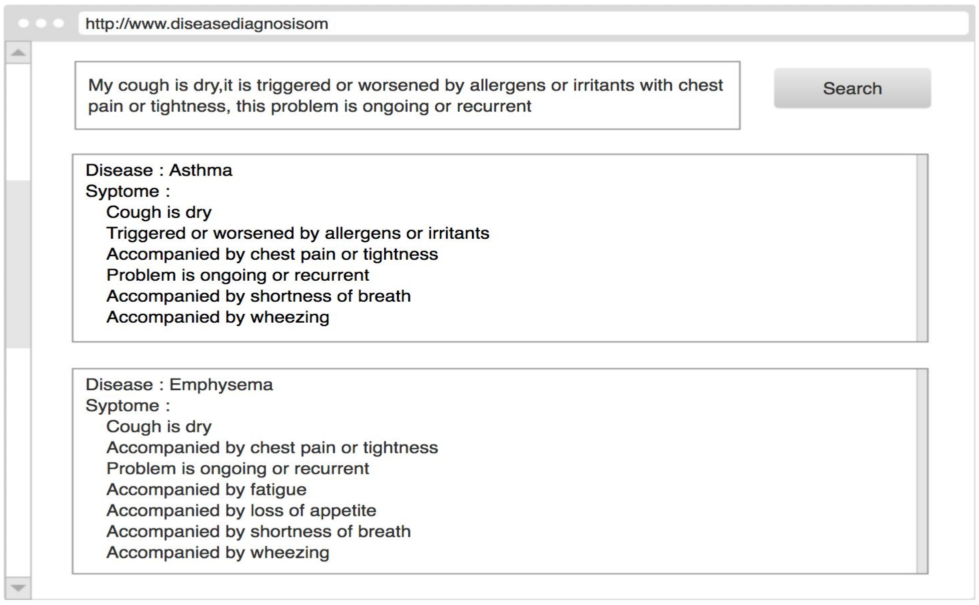
After analysing the user's input, if user's input corresponds totally our conditions (symptoms) of our rules, this is the demo of our platform :


http://www.diseasediagnosisom

I drink more, eat more and I have more urine

Disease : diabete
Syptome : drink more
eat more
urine more

But if users' input can't be find in the conditions (symptoms) of our rules, we will show all the possibilities :


http://www.diseasediagnosisom

My cough is dry,it is triggered or worsened by allergens or irritants with chest pain or tightness, this problem is ongoing or recurrent

Disease : Asthma
Syptome :
Cough is dry
Triggered or worsened by allergens or irritants
Accompanied by chest pain or tightness
Problem is ongoing or recurrent
Accompanied by shortness of breath
Accompanied by wheezing

Disease : Emphysema
Syptome :
Cough is dry
Accompanied by chest pain or tightness
Problem is ongoing or recurrent
Accompanied by fatigue
Accompanied by loss of appetite
Accompanied by shortness of breath
Accompanied by wheezing

VII - Conclusion

This project lets us realise the idea of a platform for disease diagnosis, we analyze the users' input by some semantic analysis APIs, by using the knowledge of the semantic web, we create the ontology and do the reasoning.

Compared to other platforms, we have a lot of advantages, many platforms are presented as forums and require someone to reply, in most cases, users need to read a paragraph to get the result and sometimes they may even get some irrelevant answers, there are also some platforms that require users to select their symptoms in the corresponding checkbox, this way is not convenient, because by this way, users need read the symptoms on the page line by line to do the corresponding choice. But our platform just needs users' input and then the platform calls the API to analyze the reasoning disease itself.

Through this project, we have a better understanding of the ontology and some other semantic web knowledge, we also learn some new knowledge, such as the application of SWRL in the semantic network in the domain of disease diagnosis. This project provides us a good learning opportunities for us to study in the domain of semantic web.