ISSN 2277-3061

Web-Based Knowledge Acquisition Approach for Building Diseases Symptoms Ontology

Amal AL-Harbi[1], Abdullah AL-Malaise[2], Arwa Jamjoom[3]

[1]Computer Science Department, Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

E-mail: ahsalharbi@kau.edu.sa

[2]Information Systems Department, Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

E-mail: aalmalaise@kau.edu.sa

[3]Computer Science Department, Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

E-mail: ajamjoom@kau.edu.sa

ABSTRACT

Today medical ontologies have an important role in medicine field to represent medical knowledge. They are much stronger than biomedical vocabularies. In diseases diagnosis process, each disease has number of symptoms associated with it. We can employ ontology in helping to diagnose diseases by building Diseases-Symptoms ontology, which relate diseases and symptoms. Such ontology would be very useful for medical expert systems to assist physicians in diagnosis diseases or as a training tool for medical students. In this paper, we propose a method that automatically extract medical knowledge from Web resources and build Diseases-Symptoms ontology. We use the linguistic pattern and statistical analysis techniques based on Bing search engine. We evaluated the proposed method for two diseases Hyperthyroidism and Eczema by two consultant physicians.

Keywords

Medical Ontology, Knowledge Acquisition, Linguistic Pattern, Statistical Analysis, Ontology Learning, Web mining, Ontologies.

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol.14, No.7

www.ijctonline.com, editorijctonline@gmail.com

1. INTRODUCTION

Ontologies consider as fundamental tool to represent knowledge. They consist of three main elements: classes (domain's concepts), relations (binary association between classes) and instances (individuals). Ontology is presented as an object model comprising of classes C which are taxonomically related by is-a relation $H \in C \times C$ (e.g. Eczema is-a dermatology disease) and non-taxonomically related by relations $R \in C \times C \times String$ (e.g. itchy is associated with eczema) [1]. Ontologies construction carried out by knowledge engineers and domain experts which take long time and effort. This manual approach is always described as bottleneck [2]. In this direction, ontology learning plays a crucial role in knowledge acquisition and representation process. Automated ontologies construction allows saving time and effort required by knowledge engineers and domain experts to construct specific domain ontology.

Today, Web considered as a biggest repository of information [3]. Many researchers use the Web as an effective source for knowledge acquisition task and information retrieval. We can use the Web to extract useful knowledge using ontology learning; therefore, there is a need for an unsupervised method that can ease construction of medical ontology from the Web in a cost effective manner with high accuracy.

The main objective of this paper is to build a knowledge acquisition technique that automatically extract medical knowledge related to diseases and their symptoms from Web resources and build Diseases-Symptoms ontology.

2. BACKGROUND

2.1. The Web As a Learning Corpus

In recent years, Web growth significantly and cover different domains of knowledge including a medical domain. Many classical knowledge acquisition techniques use small number of corpus, which affects the quality of extracted knowledge. Today, Web consider as the biggest repository that offer information. Using knowledge acquisition with such enormous size repository may consider a great deal.

One of the characteristics of the Web is the high redundancy of information. Authors [4] state that the relevance of information can measured by the amount of reputations between information. Web proved that it is a valid source for knowledge acquisition for many researchers in many areas include questions classification, questions answering and ontology enrichment [5].

2.2. Lightweight Analytical Approach

When using a Web as a learning corpus, the number of Web resources to be analysis is very large. To perform an efficient analysis in such case, for each Web resource we can focus on sentences contains the specifically queried concept rather than analytic the whole text. Each Web page retrieved from search engine contains at least one sentence that match the queried concept. We can then evaluate the extracted sentence to obtained relevant results.

2.3. Statistical Analysis

Statistical analysis measures techniques (co-occurrence) proved their effectiveness on finding the relatedness between concepts in unstructured data sources like the Web [4]. In proposed method, we benefit from the statistical measures, which can be immediately calculated using Web search engine hit counts. Page hit counts for query contains words (or phrases) P AND Q can considered as co-occurrence between P and Q on the Web.

2.4. Popular Co-Occurrence Measures

There are numbers of co-occurrence measures. For proposed method, we compute four popular co-occurrence measures:

2.4.1. Jaccard

Jaccard co-occurrence often used in information retrieval. It defines as:

$$WebJacard(P,Q) = \frac{H(P\cap Q)}{H(P)+H(Q)-H(P\cap Q)} \qquad (1)$$

Where $(P \cap Q)$ represent the conjunction query P AND Q and $H(P \cap Q)$ is the probability of the word P and word Q to co-occur and it represents the hit counts retrieve by search engine. This measure represents the likelihood estimate of the ratio of the probability of finding Web document that contain both words P and Q over the probability of finding a Web document contains either P or Q [6].

2.4.2. Overlap

Overlap measure is defines as:

$$WebOverlap(P,Q) = \frac{H(P\cap Q)}{\min(H(P),H(Q))} \qquad (2)$$

This measure represents the likelihood estimate of the ratio of the probability of finding Web document that contain both words P and Q over the probability of finding a Web document contains the word with a minimum occurrence [7].

2.4.3. Dice

It is very similar to the Jaccard measure and is also often used in information retrieval. It defines as:

$$WebDice(P,Q) = \frac{2 \times H(P \cap Q)}{H(P)+H(Q)} \qquad (3)$$

This measure represents the likelihood estimate of the ratio of the probability of finding Web document that contain both words P and Q over the probability of finding a Web document contains either P or Q or both [7].

2.4.4. Pointwise Mutual Information (PMI)

PMI measure is defines as [5]:

$$WebPMI(P,Q) = \log_2 \frac{H(P \cap Q)}{H(P) \times H(Q)} \qquad (4)$$

In [4], the author proposed a score measure for co-occurs between words depend on WebPMI. This score measures the relationship between two words (or noun phrases) using search engine hit counts:

$$Score(Problem, Choice_i) = \frac{H\ (Problem\ AND\ Choice_i)}{H(Choice_i)} \qquad (5)$$

Where *H* represents hit counts and *Problem* represents the problem word and $\{\, Choice_1,\ Choice_2, \ldots \ldots, Choice_n\}$ represent alternatives.

2.5. Web Search Engines

Web search engines are the tools that allow users to search information through Word Wide Web and retrieve Web documents [8]. Today we can use Web search engine to collect large number of resources. The search engine provides us a list of Web sites depend on queries.

2.5.1. Web Search Engines Classification

Web search engines classified into two types [8]:

1. Keyword-based Search Engine

Like Google, Yahoo, Bing and AltaVista. Depend on Keyword automatic algorithm to retrieve Web sites according to a user query. They provide an up to date results of Web sites. The accuracy of results depends on the user's query. They lake semantic analysis which affect the performance of such engines. If user search for a word with several meaning, the search engine cannot recognize the specific user means (e.g. word "Alahli" can be Bank name or a football team name). The large amount of retrieved Web sites is difficult to evaluate.

2. Semantic Search Engine

This meaning-based approach solves the problem of keyword-based search approach. It uses categories to organize results in a hierarchical structure. Those categories (clusters) determined by term taxonomy provided by experts.

This type can be classified into two approaches:

- Web dictionaries: such as Yahoo dictionary which contains large human classified catalogues. The user can use hierarchical structure to browse the catalogues.
- Using clustering techniques to create structure view of the results automatically. Those type of search engines provided limited number of resources if comparing with a keyword-based approach. Even they use automatic clustering techniques for Web resources, the categories are manually constructed which lead to poor semantic. They cover specific and small domains.

Many semantic search engines are no longer available because of their limitations and insufficient such as Copernic, Snaket, Kartoo and Vivisimo.

2.5.2. Web Search Engines as Learning Tools

By using a keyword-based search engine, for each query, we can retrieve a sufficient up to date set of Web resources. Also, the keyword-based search engine can be used to get statistical about information distributed and relatedness between concepts (co-occurrence).

Keyword-based search engine has some drawback including:

- All search engines allow access to limited number of Web resources even the result was millions of resources matched the query. Actually, this drawback does not affect our proposed method since we will not analysis all Web resources.
- Overhead during learning according to response time.

2.5.3. Keyword-based Search Engine Comparison

In proposed method, the search engine is a very important tool for the knowledge acquisition process. Therefore, in this section we study different available search engines and compare them in order to select the most appropriate one.

We consider the most public search engines, which are Yahoo, Google and Bing. We compare those search engines according to:

- **Access:** Some search engines allow only programmers to access their functionality by using API. Other allow only the interface and provide the result page only. In proposed method, we use API since we need to store the resulted URLs in the database.
- **Limitations:** Most search engines allow specific number of searching queries per day or per month. This to enhance the performance of the search and to avoid attack by huskers.
- **Response time:** The time needed by the search engine to provide results for the query. In proposed method, the accuracy of the result is more important than the response time. `
- **Coverage:** Number of resources the search engine index for a query. For proposed method, we will not analysis millions of pages for each query. Therefore, we do not focus on the coverage of result for a specific query.

As shown in Table 1, we performed medical domain queries through different search engines. Google offering the largest number of results and Yahoo the smallest number for medical domain queries.

Table 1 Number of Results Obtained by Several Keyword-Based

Web Search Engines for Medical Domain Queries

Query	Yahoo	Google	Bing
“Symptoms of eczema include”	8,280	**8,720**	8,290
“is associated with hypertension”	84,700	**349,000**	91,100
“Types of Diabetes”	4,320,000	**13,900,000**	4,860,000
"cancer such as"	265,000	**432,000**	292,000

In table 2, we summarise the features of each search engine. Google has a largest cover for the Web with the slowest response time. However, Google search API replaced by Customs search API and it allows free 100 queries per day. Bing has medium coverage and it is more flexible than Google in programming. It allows free 5000 queries per month without the need to embed the search box in the system. Yahoo is the most restrictive with 40 result per queries and it is not free like other search engines. For proposed method, Bing is the most appropriate search engine.

Table 2 Comparison of Most Popular Search Engines

Search Engine	Access	Limitation	Response Time	Coverage
Yahoo	API	1000 queries cost 1.2 $ Maximum 50 result for each query.	Medium	Lowest
Google	Replaced by Custom Search API[1]	Free 100 queries / day	Slowest	**Highest**
Bing	API	Free 5000 queries / month	**Fastest**	Medium

3. METHODOLOGY

The basic idea of this paper is to use ontology learning for Web-based knowledge acquisition to build Diseases-Symptoms ontology in unsupervised manner. The ontology learning from the Web is a complex process. It requires retrieve and analysis large number of unstructured resources. As mentioned in [3] and according to our search through the internet, there is no existing ontology that focuses on the relationship between diseases and their symptoms. Authors [6], proposed an alignment algorithm to align diseases ontology with the symptoms ontology manually. Since their proposed algorithm

[1] https://developers.google.com/web-search/docs/

has been designed to be performed manually; the core disease symptoms ontology they created linked a few diseases to their symptoms (11 diseases) include Diabetes type1 and type2, Anemia, Calcemia, Asthma, Adult Respiratory, Hypertension, Asthma and Rental Failure. This led to a very specific diseases symptoms ontology and that cannot be an effect in a large medical expert system that cover a broad field of the medical domain. However, authors in [3] used isolated diseases ontology DOID[2] that focus on classifying human diseases. The purpose of DODI is to provide the biomedical community with consistent, reusable and sustainable descriptions of human disease terms, phenotype characteristics and related medical vocabulary disease concepts through collaborative efforts of researchers at Northwestern University (Centre for Genetic Medicine and the University of Maryland School of Medicine, Institute for Genome Sciences). The DOID ontology last updated on 19 Nov 2014. It currently contains 8803 classes (terms) [1].

The proposed methodology is illustrated in Figure 1. As a first step, the system receives concept and PatternSet.

Concept: is a word or noun phrase that represent the domain. E.g. Eczema, Diabetes Type 1 or Breast Cancer. Those concepts will be used to construct Web queries.

PatternSet: all possible patterns text is written as a regular expression that may appear in sentences contained diseases symptoms relations (Table 3).

The system uses the concept and patternSet to construct queries e.g. "Eczema has symptoms" and "Symptoms and Signs of Eczema include".

Query: string constructed by the combination of concept and patterns in patternSet. The Query is executed in the search engine to retrieve related Web resources and to retrieve hit counts to compute statistics.

Table 3 Sample of patternSet Table

No.	Pattern Text	Example of Query
1.	* has symptoms	“Eczema has symptoms”
2.	Signs and symptoms of * include	“Signs and symptoms of Eczema include”
3.	Symptoms and Signs of * include	“Symptoms and Signs of Eczema include”
4.	Common symptoms of * include	“Common symptoms of Eczema include”

Each query executes in the search engine, is surrounding by double quotes to force search engine retrieve same matching. The retrieved sources are stored in webSet to analysis.

webSet: is the set contains retrieved Web resources according to executed query.

The main phases of proposed method include:

1. Data Pre-processing

To get a high-quality data, it should cleaned and prepared carefully before processing. This phase is an important part, which requires to implementing all data cleaning techniques appropriately; failure to do so the results will consider fake and inapplicable. As our resource for the corpus is the Web, this means we deal with a large amount of data, which requires a major effort for preparing. It is difficult to clean such amount of data. Therefore, we use a tool that can help performing this task automatically. For each returned Web source, we use Boilerplate Removal and Full-text Extraction from HTML pages. This tool takes the Web content and cleans it. It removes unrelated blocks, advertising and images.

2. Candidate Extraction

After data pre-processing phase, the cleaned content of each resource is parsed and linguistically analysis. We use natural languages processing tool to detect sentences, tokenizing and chunking the content to find the matching sentence. The system will extract the sentence that match the query. It only evaluated the nearest context of matched pattern. This allows obtaining significant results without an extensive analysis of the whole text. The system automatically extracts the noun phrases from the match sentence as candidates.

3. Candidate Selection

By this step, we have a list of extracted candidates (noun phrases). Those candidates are not necessary related to the concept and make a correct has-symptom relation. Therefore, we need to evaluate the extracted candidates and select the appropriate ones that represent the correct relation. As mentioned in section 2.4, we can benefit from statistical analysis obtained from search engine to calculate co-occurrence between concept and candidate.

[2] http://disease-ontology.org/

$$Score(Concept, Candidate) = \frac{hits\,(\text{"Concept" AND "Candidate"})}{hits(\text{Candiadate})} \qquad (6)$$

For each candidate, we calculate the score (equation 6) and compare it with the fixed threshold, if it is higher than the threshold; the extracted candidate is considered a valid symptom.

4. Data Post-processing

This phase involves evaluating the candidates and remove duplications. Before adding the newly extracted symptom for the specific disease into Diseases-Symptoms ontology, the system checks first that it does no exit.

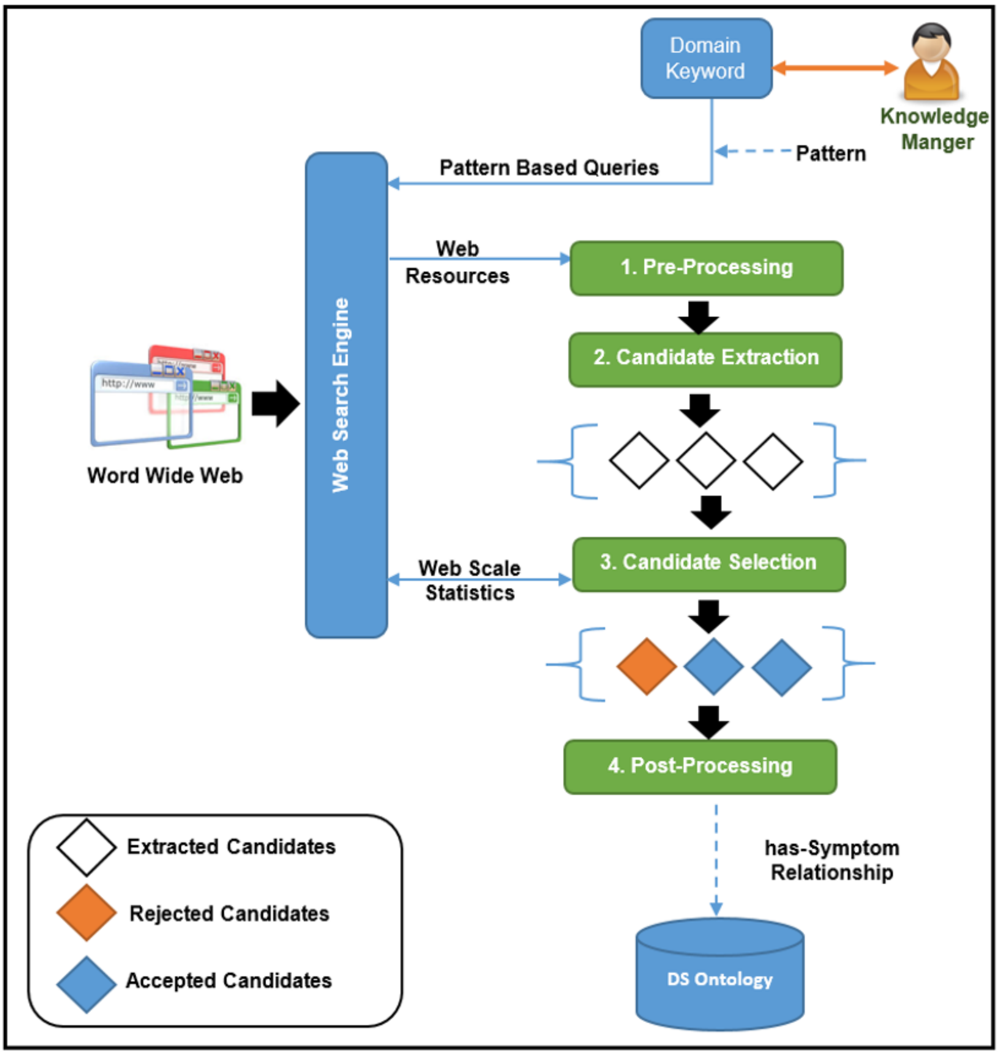


Figure 1 System Methodology

4. System Implementation

4.1. Tools and Techniques

a) Java Programming and NetBeans IDE 7.4

We chose Java as a programming language. Java is an object-oriented language, which has many features include its simplicity, high security, portability, robustly, high performance and other features [9]. NetBeans IDE is a free and open source modular developer tool for a wide range of application development technologies. It supports the development of desktop, mobile and Web applications with Java, HTML5, JavaScript, CSS and more. It has a large community of users and developers around the world [10].

b) MySQL

For system database, we chose MySQL to store the data. MySQL is the world's most popular open source database. It is a popular choice for Web applications' database and supported by NetBeans IDE [11].

c) Boilerplate Removal Tool

Boilerplate Removal and Full-text Extraction from HTML pages are a free Java tool, which can perform cleaning process in milliseconds. The boilerplate library provides algorithms to detect and remove the surplus "clutter" (boilerplate, templates) around the main textual content of a Web page. They consider that the Web content divided into two classes: long text and short text. The Long text contains the main content of the Web and the short text include the navigational text. The algorithm depends on removing the words in the short text. It achieves very high accuracy (92-98%) at almost no cost [12].

d) OWL API

The OWL API is an open source Java API. This API used to create, manipulate and serialize OWL Ontologies [13]. It allows create new ontology or modify existing ontology by adding new concepts, relations or properties.

e) Bing Search API

The Bing Search API supported by Microsoft. It allows developers to embed Web search results in applications and Websites using XML or JSON. It offers up to 5,000 free queries per month [14]. It returns for each query number of hit counts and results' URLs, titles and descriptions.

f) OpenNLP API

The OpenNLP library is a machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution. These tasks are usually required to build more advanced text processing services. OpenNLP also includes maximum entropy and perceptron based machine learning [15].

5. EVALUATION

In order to evaluate the effectiveness of the proposed system and the correctness of the constructed ontology, we conducted a pilot test. We chose two diseases from two different category: Hyperthyroidism (Thyroid Gland Disease) and Eczema (Skin Diseases). In the beginning, the database is containing only diseases without any matching symptoms. Figure 2 and Figure 3 illustrated the Diseases-Symptoms ontology after adding Hyperthyroidism symptoms.

5.1. Evaluation Measures

There are number of measures used to evaluate ontology. We apply three standard measures to evaluate proposed method for two diseases: Hyperthyroidism and Eczema. Those measures are Recall, Precision and F-Measure:

1. **Recall:** This measure shows how much of the existing knowledge extracted. It calculated by divide the number of correctly selected candidates by the total number of existing terms in related Gold Standard [16]. Since in our case there is no Gold Standard that provide the full-expected terms for the symptoms, we use consultant physician to decide the all-correct symptoms for the specific disease even if not included in the candidate list. To calculate recall, we divide the number of correctly selected candidates by the number of full set of correct symptoms decided by a consultant physician.

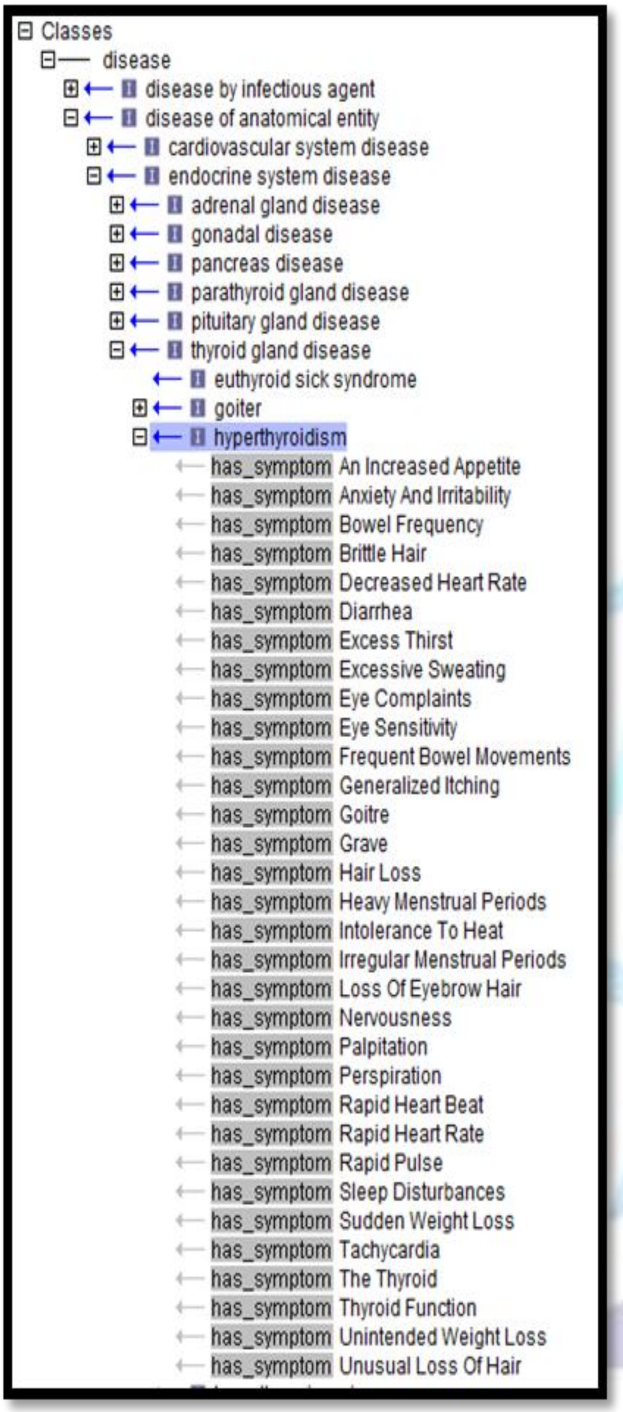
$$Recall = \frac{no.of\ correctly\ selected\ candidates}{no.of\ full\ set\ of\ correct\ symptoms} \qquad (7)$$

2. **Precision:** This measure states to which degree the knowledge is extracted correctly. It represents the ratio between the number of correctly selected candidates and the total number of extracted candidates [16].

$$Precision = \frac{no.of\ correctly\ selected\ candidates}{no.of\ extracted\ candidates} \qquad (8)$$

3. **F-Measure:** This measure provides the weighted harmonic mean of Precision and Recall [16].

$$F\_Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (9)$$



- ⊟ Classes
  - ⊟ disease
    - ⊞ disease by infectious agent
    - ⊟ disease of anatomical entity
      - ⊞ cardiovascular system disease
      - ⊟ endocrine system disease
        - ⊞ adrenal gland disease
        - ⊞ gonadal disease
        - ⊞ pancreas disease
        - ⊞ parathyroid gland disease
        - ⊞ pituitary gland disease
        - ⊟ thyroid gland disease
          - euthyroid sick syndrome
          - ⊞ goiter
          - ⊟ hyperthyroidism
            - has_symptom An Increased Appetite
            - has_symptom Anxiety And Irritability
            - has_symptom Bowel Frequency
            - has_symptom Brittle Hair
            - has_symptom Decreased Heart Rate
            - has_symptom Diarrhea
            - has_symptom Excess Thirst
            - has_symptom Excessive Sweating
            - has_symptom Eye Complaints
            - has_symptom Eye Sensitivity
            - has_symptom Frequent Bowel Movements
            - has_symptom Generalized Itching
            - has_symptom Goitre
            - has_symptom Grave
            - has_symptom Hair Loss
            - has_symptom Heavy Menstrual Periods
            - has_symptom Intolerance To Heat
            - has_symptom Irregular Menstrual Periods
            - has_symptom Loss Of Eyebrow Hair
            - has_symptom Nervousness
            - has_symptom Palpitation
            - has_symptom Perspiration
            - has_symptom Rapid Heart Beat
            - has_symptom Rapid Heart Rate
            - has_symptom Rapid Pulse
            - has_symptom Sleep Disturbances
            - has_symptom Sudden Weight Loss
            - has_symptom Tachycardia
            - has_symptom The Thyroid
            - has_symptom Thyroid Function
            - has_symptom Unintended Weight Loss
            - has_symptom Unusual Loss Of Hair

Figure 2 Diseases-Symptoms Ontology has-Symptom relation

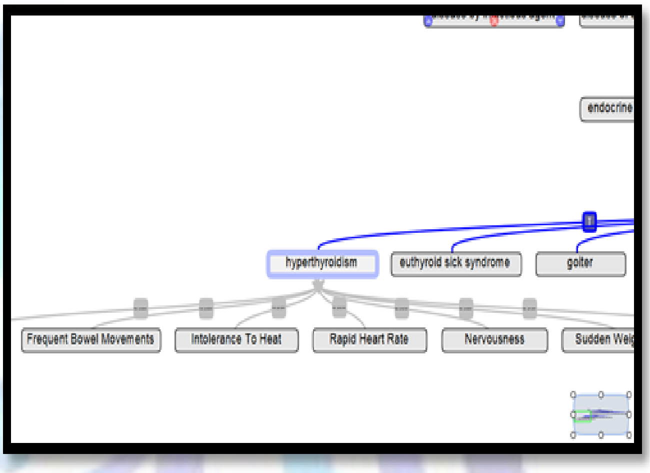


Figure 3 Diseases-Symptoms Ontology Hyperthyroidism's Symptoms

5.2. Evaluation Procedure

For evaluation, we presented the output to consultant physicians to decide which output is accepted as symptom and what are the symptoms that not extracted by our system. For Hyperthyroidism disease, we presented the output to Dr. Abdulqawi Almansari (Head of Endocrinology and Diabetes Centre at Bagado and Dr. Erfan Hospital - Jeddah). The number of symptoms decides by the physician is 38 symptoms. For Eczema disease, we presented the outputs to Dr. Faiza Al-Tajem (Dermatology Consultant at King Abdulaziz Medical City - Jeddah). The number of symptoms decides by the physician is 50 symptoms with five symptoms added by her (not extracted by the system).

Initially, we test different values of the threshold to decide which value is more appropriate for the medical domain. We evaluate three different threshold values 0.1, 0.15 and 0.2. Results summarized in table 4 for Hyperthyroidism and table 5 for Eczema.

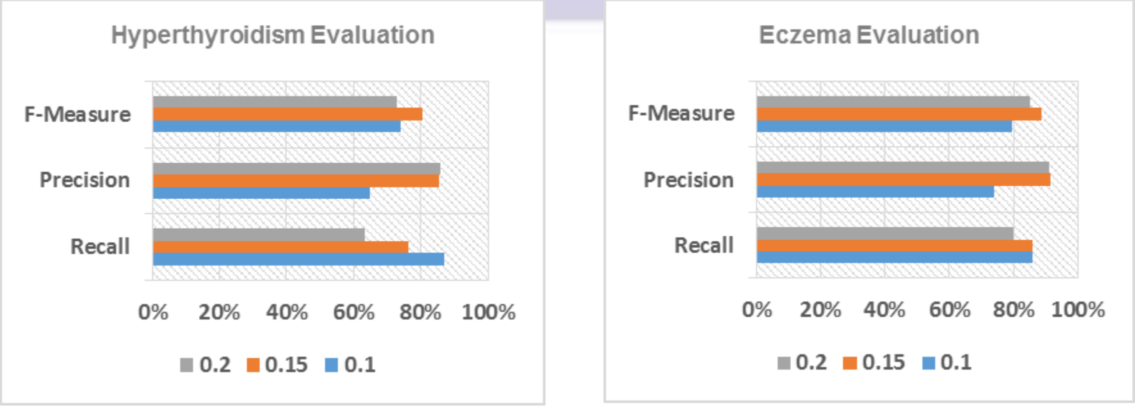
Table 4 Number of Correctly and Incorrectly Selected and Rejected Candidates for Hyperthyroidism. (Total Number of Correct Symptoms 38)

		Right	Wrong	Total
Threshold = 0.1	**Accepted**	24	18	**51**
	Rejected	113	5	**104**
	Total	**137**	**23**	**155**
Threshold = 0.15	**Accepted**	29	5	**34**
	Rejected	112	9	**121**
	Total	**141**	**14**	**155**
Threshold = 0.2	**Accepted**	24	4	**28**
	Rejected	113	14	**127**
	Total	**137**	**18**	**155**

Table 5 Number of Correctly and Incorrectly Selected and Rejected Candidates for Eczema. (Total Number of correct symptoms 50)

		Right	Wrong	Total
Threshold = 0.1	**Accepted**	43	15	**58**
	Rejected	102	2	**104**
	Total	**145**	**17**	**162**
Threshold = 0.15	**Accepted**	43	4	**47**
	Rejected	113	2	**115**
	Total	**156**	**6**	**162**
Threshold = 0.2	**Accepted**	40	4	**44**
	Rejected	113	5	**118**
	Total	**153**	**9**	**162**

Observing the results, we can see that, number of correctly selected and rejected candidates is higher than the number of mistakes (Incorrectly selected and rejected candidates). From Figure 4(a) and Figure 4(b), we can conclude that the F-measure achieve the best value when the threshold is 0.15.



(a) Hyperthyroidism

(b) Eczema

Figure 4 Comparison of Different Threshold Values for (a) Hyperthyroidism (b) Eczema

We also evaluate the number of extracted candidates according to different number of Web resources. For Hyperthyroidism, total number of Web resources is 104 since some queries return few numbers of results. As we can see in Figure 5, when the number of Web resources increased the number of extracted candidates is also increased. With 100 Web resources, number of extracted candidates without duplication is 201 candidates.

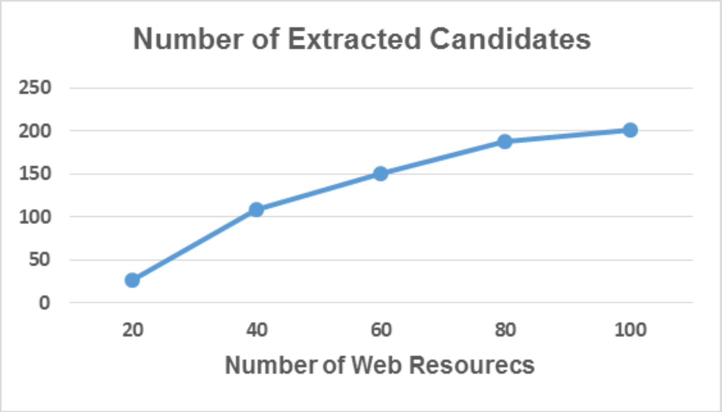
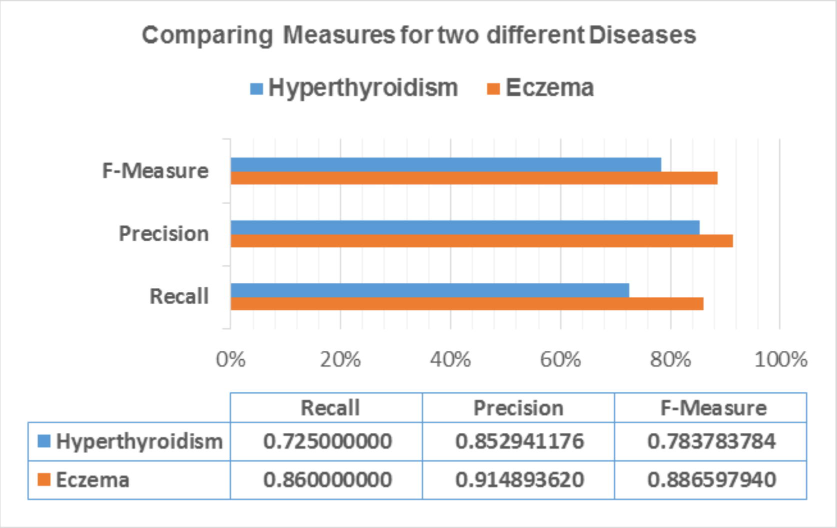


Figure 5 Number of Extracted Candidates for Hyperthyroidism

6. RESULTS

We can conclude that with 0.15 threshold and maximum number of Web resources provided by Bing API, the results of two different diseases (Hyperthyroidism and Eczema) are very good (Figure 6). In the medical context, recall is moreover regarded as primary measure, as the aim is to identify all correct cases. For Hyperthyroidism, 72% of correct symptoms extracted by our proposed system. For eczema, 86% of correct symptoms extracted. The F-measure achieve high value in both cases, 78% and 88% for Hyperthyroidism and Eczema respectively. Applying our method for any other diseases will lead to the same very good results.



	Recall	Precision	F-Measure
Hyperthyroidism	0.725000000	0.852941176	0.783783784
Eczema	0.860000000	0.914893620	0.886597940

Figure 6 Evaluation of the Performance of Our Proposed System

For Hyperthyroidism and Eczema.

7. CONCLUSION AND FUTURE WORK

Medical ontologies have an important role in medicine field. We can employ ontology in helping to diagnose diseases by building Diseases-Symptoms ontology, which relate diseases and symptoms. In this paper, we proposed an automatic and unsupervised method to acquire medical knowledge related to diseases and their symptoms from the Web. The main result of proposed method is a Diseases-Symptoms ontology, which contains a hierarchy of human diseases and their symptoms. We used linguistic pattern and statistical analysis techniques based on Bing search engine. We evaluate our proposed system for two diseases Hyperthyroidism and Eczema by two consultant physicians.

The study yielded that with 0.15 threshold and maximum number of Web resources provided by Bing API, the results were very good. The Number of correctly selected and rejected symptoms for a specific disease is higher than the number of mistakes (Incorrectly selected and rejected candidates). The resulted Diseases Symptom ontology would be very useful for medical expert systems to assist physicians in diagnosis diseases or as a training tool for medical students.

As future work, we may focus on analysis medical Web sites rather than whole Web. We may use snippet provided by search engine to extract candidate rather than download Web resource content. We also may include the finding, laboratory results and medicine for each disease.

ACKNOWLEDGMENT

First author would like to acknowledge and thank Dr. Abdulqawi Almansari (Head of Endocrinology and Diabetes Centre at Bagado and Dr. Erfan Hospital - Jeddah) and Dr. Faiza Al-Tajem (Dermatology Consultant at King Abdulaziz Medical City - Jeddah) for their contribution towards evaluate the system.

REFERENCES

[1] K. WA, A. C, F. V, M. E, B. E, F. G, M. CJ, B. JX, M. J, V. D, P. H and S. LM, “Disease Ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data,” *Nucleic Acids Research,* pp. 1-8, October 2014.

[2] C. Wagne, “Breaking the Knowledge Acquisition Bottleneck Through Conversational Knowledge Management,” *70 Information Resources Management Journal,* pp. 70-72, 2006.

[3] O. Mohammed, R. Benlamri and S. Fong, “Building a Diseases Symptoms Ontology for Medical Diagnosis: An Integrative Approach,” in *Future Generation Communication Technology (FGCT)*, London, 2012.

[4] D. Sanchez and A. Moreno, “Learning non-taxonomic relationships from web documents for domain ontology construction,” *International Journal of Data & Knowledge Engineering 64,* pp. 600-623, 2008.

[5] P. D. Turney, “Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL,” in *In Proceedings of the 12th European Conference on Machine Learning* , London, 2001.

[6] G. Salton and M. J. McGill, in *Introduction to Modern Information Retrieval.*, New York, McGraw-Hill, Inc, 1983.

[7] D. Bollegala, Y. Matsuo and M. Ishizuka, “A Web Search Engine-Based Approach to Measure Semantic Similarity between Words,” *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,* vol. 23, pp. 977-990, July 2011.

[8] J. Singh and A. Sharan, “A Comparative Study between Keyword and Semantic Based Search Engines,” in *International Conference on Cloud*, 2013.

[9] in *Java The Complete Reference, 9th Edition*, Oracle Press, March 11, 2014.

[10] Oracle, “NetBeans IDE Features,” 2015. [Online]. Available: https://netbeans.org/features/ide/index.html.

[11] Oracle, 2015. [Online]. Available: http://www.mysql.com/.

[12] C. Kohlschütter, P. Fankhauser and W. Nejdl, “Boilerplate detection using shallow text features,” in *Proceedings of the third ACM international conference on Web search and data mining*, New York, 2010.

[13] GitHub, “The OWL API,” 2015. [Online]. Available: http://owlapi.sourceforge.net/.

[14] “The Bing Search API is now available on the Windows Azure Marketplace,” Microsoft, 2015. [Online]. Available: http://www.bing.com/toolbox/bingsearchapi.

[15] “Welcome to Apache OpenNLP,” 2010. [Online]. Available: https://opennlp.apache.org/index.html.

[16] D. M. W. Powers, “EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION,” *Journal of Machine Learning Technologies,* pp. 37-63, 2011.