35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Estimating Personalized Risk Ranking using Laboratory Test and Medical Knowledge (UMLS)

Meru A Patil*,1, Sandip Bhaumik[1], Soubhik Paul[1], Swarupananda Bissoyi[1], Raj Roy[1], Seungwoo Ryu[2]

Abstract— **In this paper, we introduce a Concept Graph Engine (CG-Engine) that generates patient specific personalized disease ranking based on the laboratory test data. CG-Engine uses the Unified Medical Language System database as medical knowledge base. The CG-Engine consists of two concepts namely, a concept graph and its attributes. The concept graph is a two level tree that starts at a laboratory test root node and ends at a disease node. The attributes of concept graph are: Relation types, Semantic types, Number of Sources and Symmetric Information between nodes. These attributes are used to compute the weight between laboratory tests and diseases. The personalized disease ranking is created by aggregating the weights of all the paths connecting between a particular disease and contributing abnormal laboratory tests. The clinical application of CG-Engine improves physician's throughput as it provides the snapshot view of abnormal laboratory tests as well as a personalized disease ranking.**

Index Terms— **Personalized Risk, Concept Graph, Laboratory test, Disease Ranking, UMLS.**

I. INTRODUCTION

The laboratory tests play an important role in a clinical scenario for screening and/or diagnosis of diseases. It was observed by Mindemark *et al.* [1] that in a 7-year period (2002-2008), the laboratory test usage in a hospital setup has increased by 70% and the number of available laboratory tests have increased by over 140%. Often in a routine annual health check-up, close to 100 laboratory tests are performed per patient. Majority of these tests are generic in nature as they do not target any specific disease. However, they do provide vital clues about the presence or absence of diseases. Hence, physicians widely use these tests routinely. Currently the physician-to-patient ratio in countries like India hovers at around 6 per 10000 people [2]. This skewed physician-to-patient ratio along with an increased usage of laboratory tests has resulted in a physician spending majority of their consultation time on the analysis of laboratory reports.

There were efforts made by Bauer *et al.* [3] to enhance a physician's throughput by the means of an alternative visualization of the laboratory reports. In a study done by Torsvik *et al.* [4], it was observed that alternative visualizations of laboratory test reports helped clinicians in few special cases. However, it was also observed that these techniques are not ideal in general scenarios. These alternative visualizations fail to increase a physician's throughput as they lack a comprehensive report analysis. Hence, an automated system that can analyse laboratory test reports and identify associated disease risks will greatly help in increasing a physician's throughput.

In this work, we introduce such an automated system called as Concept Graph Engine (CG-Engine) that is based on Unified Medical Language System (UMLS). The CG-Engine takes laboratory test data as its input and generates probable disease risks by analysing abnormal test results. UMLS is a knowledge representation framework that includes more than 100 medical terminology sources. There are three knowledge sources (databases) in UMLS, namely: Metathesaurus, Semantic Network and SPECIALIST Lexicon. Metathesaurus along with Semantic Network is used by Caviedes *et al.* [5] in finding the similarity between two concepts of the UMLS. McInnes *et al.* [6] developed a software package based on this work. Dupuch *et al.* [7] exploited parameters of UMLS to find similarity between two concepts. Co-occurrence information of the concepts stored in MRCOC table of UMLS was used by Zeng *et al.* [8] to study the sensitivities of disease-drug chemical relationship and disease-lab chemical relationships. Volot *et al.* [9] created Concept Type Lattice (CTL) using UMLS as a source to acquire medical knowledge. Relationship information between concepts defined in MRREL table of UMLS was used by Liu *et al.* [10] for resolution of the ambiguity between terms. The difference between the above listed studies and our work is that in addition to using these knowledge sources, we use the various attributes of UMLS to construct the "Laboratory Test-Disease" (LT-D) relationship.

This article is organized as follows. In Section II, we give high level view of CG-Engine. The building of the graph from the laboratory test to disease is developed in Section II-A. The relevance of a particular test to a disease is measured by some weights. These weights and their computation are described in Section II-B. In Section III results of CG-Engine are discussed with respect to liver diseases and diabetes mellitus. Finally in Section IV we conclude the discussion with a summary and the future scope of this work.

II. METHODOLOGY

The proposed CG Engine is based on concept graph between laboratory test to diseases and the weight parameters of the graph. The CG Engine is built as a two-step process; below Subsections explain both the steps of the engine.

A. Building of Laboratory Test-Disease (LT-D) concept graph

Building of LT-D concept graph involves creation of a new database (DB) schema called LT-D DB out of UMLS

* Email:meru.patil@samsung.com
[1] Affiliation : Samsung Advanced Institute of Technology, India.
[2] Affiliation : Samsung Advanced Institute of Technology, Korea.

978-1-4577-0216-7/13/$26.00 ©2013 IEEE

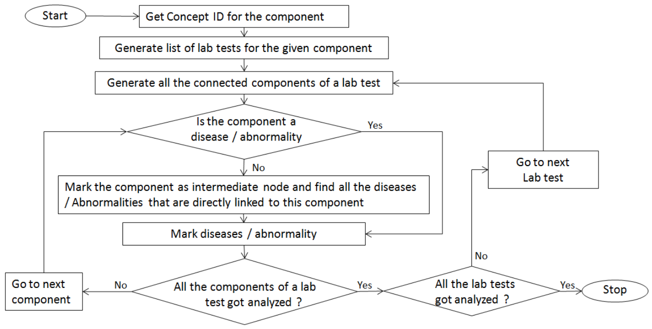

Start
Get Concept ID for the component
Generate list of lab tests for the given component
Generate all the connected components of a lab test
Is the component a disease / abnormality
Yes
No
Mark the component as intermediate node and find all the diseases / Abnormalities that are directly linked to this component
Mark diseases / abnormality
Go to next Lab test
Go to next component
No
All the components of a lab test got analyzed ?
Yes
No
All the lab tests got analyzed ?
Yes
Stop

Fig. 1. Concept Graph Generation Flow chart.

DB (version 2012AA). The first step of creating LT-D DB involves finding Concept Unique Identifier (CUI) of the component in UMLS DB. A component CUI is one that is associated with clinical laboratory test. For example "*Cholesterol*" is the associated component for "*Blood Cholesterol*" clinical laboratory test and its CUI is C0008377. A query is executed on the UMLS DB to find all the entries (called UMLS laboratory tests) that have semantic type as "*Laboratory Procedure*" and are directly related to the given component (see Fig. 1). The UMLS laboratory tests that actually represent the underlying clinical test are considered and remaining entries are discarded. For example, the following UMLS laboratory tests are found for "*Blood Cholesterol*": '*Serum cholesterol measurement*', '*Cholesterol measurement test*', '*isolation & purification analysis*', '*Plasma Cholesterol Test*', '*Serum total cholesterol measurement*' and '*serum LDL*'. Among them '*isolation & purification analysis*' and '*serum LDL*' laboratory tests are removed from the mapping list as they are unrelated to this clinical laboratory test. In a few of the cases, the UMLS DB query did not yield good result. In such cases, Metathesaurus search of UMLS was used to find relevant UMLS laboratory test entries. The degree of relevance for a given UMLS laboratory test to a particular clinical test is validated by using online resource [11]. The above method is used on 100+ clinical laboratory tests and each clinical test is mapped to one or more UMLS laboratory tests.

The next step is to build a LT-D concept graph, which involves querying the DB to find all the connected entries of the UMLS laboratory test. Semantic type of each entry is checked to classify it as either a disease node or an intermediate node of the graph. For example if an entry has semantic type as "*Pathologic Function*" then it is considered as a disease node whereas if it has "*Lipid*" semantic type then it is considered as an intermediate node. Out of 133 semantic types, 5 types are considered for disease nodes and 62 types for intermediate nodes. Recursive querying is performed on the intermediate nodes to find all the directly connected disease nodes (see Fig. 1). While building the graph, the following path attributes are stored (see Fig. 2) along with semantic type of nodes: Symmetric information, Number of sources and Relation types. All the 11 valid relation types present in UMLS 2012AA DB are considered.

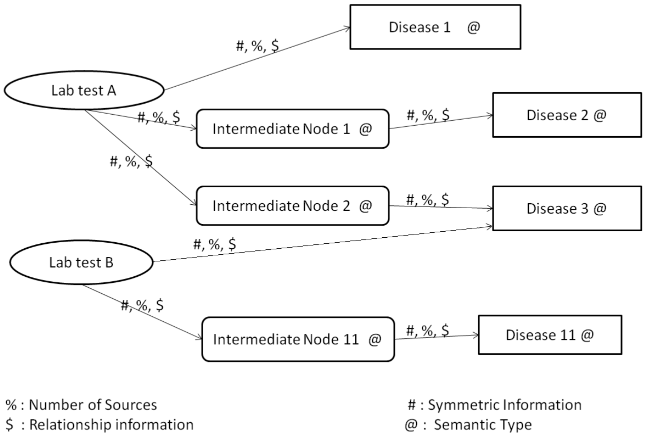


Fig. 2. Concept graph with all the attributes.

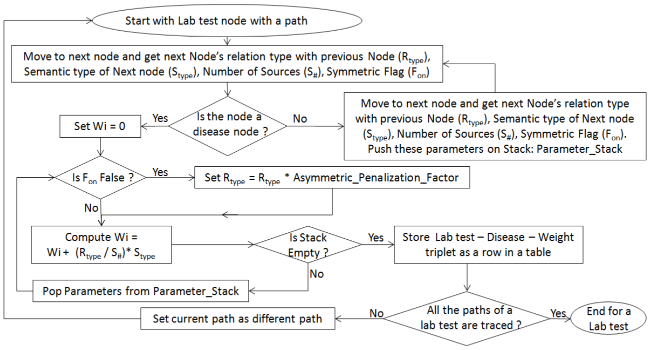

Start with Lab test node with a path
Move to next node and get next Node's relation type with previous Node (R_type), Semantic type of Next node (S_type), Number of Sources (S_#), Symmetric Flag (F_on)
Is the node a disease node ?
Yes
Set Wi = 0
No
Move to next node and get next Node's relation type with previous Node (R_type), Semantic type of Next node (S_type), Number of Sources (S_#), Symmetric Flag (F_on). Push these parameters on Stack: Parameter_Stack
Is F_on False ?
Yes
Set R_type = R_type * Asymmetric_Penalization_Factor
No
Compute Wi = Wi + (R_type / S_#)* S_type
Is Stack Empty ?
Yes
Store Lab test – Disease – Weight triplet as a row in a table
No
Pop Parameters from Parameter_Stack
All the paths of a lab test are traced ?
No
Set current path as different path
Yes
End for a Lab test

Fig. 3. Laboratory Test-Disease pair weight computation flow chart.

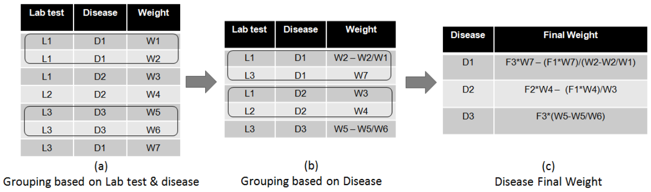
B. Computation of Weight

In a concept graph, the relevance of a particular laboratory test to a disease is measured by weighing its path to the disease. Each attribute of the concept graph is heuristically assigned a weight, for example: "*Narrower Relationship*" between laboratory test to intermediate node is given weight of 1; "*Clinical Attribute*" semantic type is given weight of 19. The lower weight is used to signify closer association and vice-versa. The weight for "Number of Source" attribute is equivalent to its value. Similarly, symmetric information attribute has weight of 1 if *true*, else it is equivalent to "*Asymmetric Penalization Factor*". Based on these values, the weight $W_{N_1-N_2}$ between two nodes is calculated as (see Eq. (1)):

$$W_{N_1-N_2} = \frac{R_{\text{type}}}{S_{\#}} \times S_{\text{type}} \qquad (1)$$

where R_{type} represents Relationship weight (if symmetric information is *true*) or Relationship weight × Asymmetric Penalization Factor (if symmetric information is *false*). $S_{\#}$ is the Number of sources and S_{type} is the Semantic type weight.

The weight for LT-D pair for the path: Laboratory Test node-Intermediate node-Disease node is the addition of weights between the Laboratory Test node-Intermediate node and Intermediate node-Disease node. Fig. 3 depicts the flowchart for LT-D pair weight computation method.



Lab test	Disease	Weight
L1	D1	W1
L1	D1	W2
L1	D2	W3
L2	D2	W4
L3	D3	W5
L3	D3	W6
L3	D1	W7

(a)
Grouping based on Lab test & disease

Lab test	Disease	Weight
L1	D1	W2 – W2/W1
L3	D1	W7
L1	D2	W3
L2	D2	W4
L3	D3	W5 – W5/W6

(b)
Grouping based on Disease

Disease	Final Weight
D1	F3*W7 – (F1*W7)/(W2-W2/W1)
D2	F2*W4 – (F1*W4)/W3
D3	F3*(W5-W5/W6)

(c)
Disease Final Weight

Fig. 4. Aggregation of weights: (a) Weights aggregated based on laboratory test and Disease: $W_2 < W_1$ and $W_5 < W_6$. (b) Weights aggregated based on disease: $W_7 < (W_2 - W_2/W_1), W_4 < W_3$ and $W_5 < W_6$. (c) Weight formula for each disease

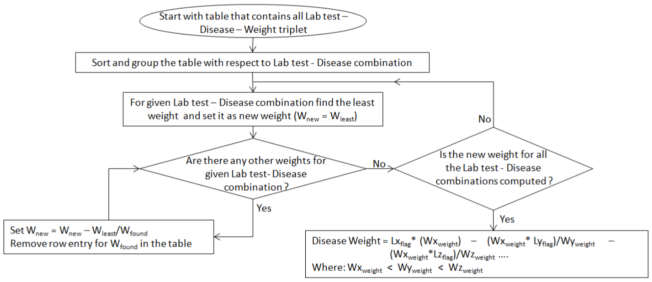

Start with table that contains all Lab test – Disease – Weight triplet
Sort and group the table with respect to Lab test - Disease combination
For given Lab test – Disease combination find the least weight and set it as new weight (W_{new} = W_{least})
Are there any other weights for given Lab test- Disease combination ?
No
Is the new weight for all the Lab test - Disease combinations computed ?
No
Yes
Set W_{new} = W_{new} – W_{least}/W_{found}
Remove row entry for W_{found} in the table
Yes
Disease Weight = Lx_{flag}* (Wx_{weight}) – (Wx_{weight}* Ly_{flag})/Wy_{weight} – (Wx_{weight}*Lz_{flag})/Wz_{weight} ….
Where: Wx_{weight} < Wy_{weight} < Wz_{weight}

Fig. 5. Disease weight computation flow chart.

The LT-D pair weights computed in above steps are for each path of laboratory test to disease in the concept graph. However, a laboratory test can be connected to disease via multiple paths. Say a laboratory test L_1 is connected to disease D_1 via two paths namely: Path 1 and 2 with their respective weights as W_1 and W_2 (see Fig. 4(a)). The aggregated weight between laboratory test L_1 and disease D_1 is $W_2 - W_2/W_1$ with an assumption that $W_2 < W_1$ (see Fig. 4(b)). For a laboratory test having N paths to a disease, the weight is calculated as (see Eq. (2)):

$$\gamma = \alpha \left(1 - \sum_{i=2}^{N} \frac{1}{\beta_i} \right) \tag{2}$$

where γ represents aggregated weight between laboratory test and disease. α is the least path weight between laboratory test and disease. β_i are the weights of the remaining $N - 1$ paths.

Finally, a disease weight is computed by aggregating weights of all the contributing laboratory tests. This is in similar lines as the computation of the LT-D pair weight (see Eq. (2)). For a disease that is screened using M laboratory tests, the weight is calculated as (see Eq. (3)):

$$\delta = \mu \left(1 - \sum_{j=2}^{M} \frac{1}{\lambda j} \right) \tag{3}$$

where δ is the disease weight. μ is the least weight of LT-D pair. λj are the weights of remaining $M - 1$ LT-D pairs. Fig. 4(b) and 4(c) shows the aggregation of weights for disease D_1, D_2 and D_3.

The above method of disease weights computation considers all the related laboratory tests of a disease. However, it is a known fact that clinically only abnormal laboratory tests are used to diagnosis/screen a disease. Hence, it is important to consider only the abnormal laboratory test weights. This is achieved by introduction of binary flags to each laboratory tests. Say laboratory test L_1 is represented by a binary flag F_1. The flag F_1 will be *true* only when laboratory test L_1 is abnormal, else it will be *false* (see Fig. 4(c)). This method of disease weight computation will work in all cases except when the laboratory test with least weight is normal. In such cases, the abnormal laboratory test with the next least weight is considered as least weight and the computation is performed. Fig. 5 illustrates the flow chart for disease weight computation using weights of LT-D pair.

III. Results and Discussion

The CG-Engine is verified on a wide set of clinical cases; the engine was run on 441 cases taken from a hospital DB. Each of the cases has information that can be broadly classified into 3 categories: laboratory test data, non-laboratory test data like radiology reports and a set of clinician's annotations. The clinician's annotations cover wide range of information from radiology reports to diet planning. In our experiments, CG-Engine results are verified for liver disease and diabetes mellitus against available doctor's annotation. Below Subsections explain the verification methodology and CG-Engine's sensitivity for these two diseases.

A. *Verification of CG-Engine for Liver diseases*

Among the 441 cases reviewed, 149 cases have annotations that are specific to fatty liver. Out of 149 cases, 112 cases are annotated based on ultrasonography report data and in the rest of the cases, the annotations do not mention any diagnostic methodology. However, the annotations capture the severity level in all the 149 cases, they are categorized into the following four types: Minimal Fatty Liver, Mild Fatty Liver, Moderate Fatty Liver and Severe Fatty Liver.

The CG-Engine was run on all the 149 cases, as all the cases had their corresponding laboratory test data. The CG-Engine provides the output in the form of the list of abnormalities with their relative ranks for the given set of abnormal laboratory test data. The list of all possible abnormalities that can be given by the CG-Engine are reviewed. From this list following 3 liver diseases are considered for monitoring: '*Disease of Liver & Biliary System (C*0267792*)*', '*Liver Dysfunction (C*0086565*)*' and '*Liver Diseases (C*0023895*)*'. For each case, the CG-Engine output is classified into 5 rank categories. The ranks of the above listed 3 diseases are analyzed, the lowest rank among them is considered and the counter in corresponding rank category is increased. Table I shows summary of the CG-Engine result for all 149 cases.

The Table I shows that in 99 cases, one of the above 3 considered liver diseases are listed in top 10 rank by the CG-Engine. Whereas, in 77.9% of cases one of them is listed in top 20 by the CG-Engine. For the 112 cases that are annotated using ultrasonography, CG-Engine listed one of the above diseases into top 20 ranks for 88 cases (78.6%).

TABLE I

SUMMARY OF CG-ENGINE RESULT FOR LIVER DISEASES CASES. THE VALUES IN BRACES ARE NUMBER OF CASES

Rank	**Minimal Fatty Liver** %(41)	**Mild Fatty Liver** %(50)	**Moderate Fatty Liver** %(54)	**Severe Fatty Liver** %(4)	**Total** %(149)
Top 10	58.5 (24)	60 (30)	75.9 (41)	100 (4)	66.4 (99)
Top 20	75.6 (31)	70 (35)	85.2 (46)	100 (4)	77.9 (116)
Top 30	75.6 (31)	76 (38)	90.7 (49)	100 (4)	81.9 (122)
Top 40	75.6 (31)	80 (40)	90.7 (49)	100 (4)	83.2 (124)
All	97.6 (40)	100 (50)	98.2 (53)	100 (4)	98.7 (147)

In 37 non-specific annotation cases, the top 20 rank hit rate is 75.7%.

It is interesting to observe from Table I that in 75.6% of Minimal Fatty Liver cases the CG-Engine could successfully list one of the above considered diseases in top 20 ranks. Thus, the CG-Engine shows very high sensitivity, considering the fact that Minimal Fatty Liver cases are very difficult to diagnose by clinical examination or by ultrasonography tests.

B. Verification of CG-Engine for Diabetes Mellitus

The CG-Engine is also verified for estimating the occurrences of Diabetes Mellitus using laboratory test. The procedure used in verification in this case is similar to that followed for verification of Liver Diseases. 166 cases out of 441 have Diabetes related clinician's comments. Unlike Fatty Liver comments, these comments do not mention the severity of the abnormality. So, no severity-based classification was performed. The following two diseases are considered for monitoring: '*Hyperglycemia (C*0020456*)*' and '*Diabetes Mellitus (C*0011849*)*'. For each case, the CG-Engine output is classified into 5 rank categories. The ranks of the above listed two diseases are analyzed, the lowest rank among them is considered and the counter in the corresponding rank category is increased. Table II shows the summary of the CG-Engine result for all the 166 cases.

TABLE II

SUMMARY OF CG-ENGINE RESULT FOR 166 DIABETES CASES. THE VALUES IN BRACES CORRESPOND TO NUMBER OF CASES

Rank	**Hyperglycemia** %	**Diabetes Mellitus** %	**Hyperglycemia OR Diabetes Mellitus** %
Top 10	47.6 (79)	4.2 (7)	48.8 (81)
Top 20	61.4 (102)	16.9 (28)	65.1 (108)
Top 30	68.7 (114)	31.3 (52)	74.7 (124)
Top 40	77.1 (128)	46.4 (77)	88 (146)
All	99.4 (165)	99.4 (165)	99.4 (165)

The Table II shows that in 81 cases, one of the above two considered diseases are listed in top 10 rank by CG-Engine. Whereas, in 65.1% of cases one of them is listed in top 20 by CG-Engine. The relative lower success rate of CG-Engine in predicting Diabetes Mellitus compared to Liver Disease risk can be attributed to following reasons:

- Few cases where annotations like: "*Please maintain the anti-diabetic management that you have got already*" are considered as diabetes case, whereas in reality this case may be pre-diabetes case with normal laboratory test outcomes.
- Diabetes has many primary and secondary complications. Hence, all the 166 cases also have other diseases like liver diseases and hypertension. These diseases have occupied the top slots of CG-Engine output as compared to Hyperglycemia or Diabetes Mellitus. This trend can be seen in the above tables as there is 9.6% increase for top 20 to top 30 for diabetic cases, whereas it is just 4% increase for same range in Liver Disease cases.

IV. CONCLUSION AND FUTURE WORKS

In overall, the CG-Engine is an effective method in summarizing laboratory test results, giving an output as a list of ranked risks for a given list of laboratory tests. The CG-Engine can be effectively used as the first level screening mechanism for wide range of clinical abnormalities and based on its outcome further diagnostic steps can be taken by the clinicians. This paper discusses the CG-Engine development for finding risks based on laboratory tests. However, the same work can be extended to find out list of most suitable diagnostic or treatment procedures for given set of abnormalities.

REFERENCES

[1] M. Mindemark and A. Larsson, Longitudinal trends in laboratory test utilization at a large tertiary care university hospital in Sweden, Upsala J. Med. Sci., vol. 116, pp. 34-38, 2011.

[2] World Bank Data, accessed on December 30, 2012, http://data.worldbank.org/indicator/SH.MED.PHYS.ZS

[3] D. T. Bauer, S. Guerlain, and P. J. Brown, The design and evaluation of a graphical display for laboratory data, J. Am. Med. Inform. Assoc., vol. 17, pp. 416-424, 2010.

[4] T. Torsvik, B. Lillebo, and G. Mikkelsen, Presentation of clinical laboratory results: an experimental comparison of four visualization techniques, J. Am. Med. Inform. Assoc., vol. 20, pp. 325-331, 2012.

[5] J. E. Caviedes and J. J. Cimino, Towards the development of a conceptual distance metric for the UMLS, J. Biomed. Inform., vol. 37, pp. 77-85, 2004.

[6] B. T. McInnes, T. Pedersen, and S. V. S. Pakhomov, UMLS-Interface and UMLS-Similarity : Open Source Software for Measuring Paths and Semantic Similarity, AMIA Annu. Symp. Proc., pp. 431-435, 2009.

[7] M. Dupuch, L. Trinquart, I. Colombet, M. Jaulent, and N. Grabar, Exploitation of semantic similarity for adaptation of existing terminologies within biomedical area, in Proc. International Conference on Knowledge Engineering Knowledge Management (EKAW), 2010.

[8] Q. Zeng and J. J. Cimino, Automated Knowledge Extraction from the UMLS, in Proc. AMIA Symp., pp. 568-572, 1998.

[9] F. Volot, P. Zweigenbaum, B. Bachimont, M. B. Said, J. Bouaud, M. Fieschi, and J. F. Boisvieux, Structuration and Acquisition of Medical Knowledge Using UMLS in the Conceptual Graph Formalism, in Proc. Annu. Symp. Comput. Appl. Med. Care, pp. 710-714, 1993.

[10] H. Liu, S. B. Johnson, and C. Friedman, Automatic Resolution of Ambiguous Terms Based on Machine Learning and Conceptual Relations in the UMLS, J. Am. Med. Inform. Assoc., vol. 9, pp. 621-636, 2002.

[11] Lab Tests Online, accessed on December 25, 2012, http://labtestsonline.org/