

# TaxaMiner: Improving Taxonomy Label Quality Using Latent Semantic Indexing

Cartic Ramakrishnan<sup>1</sup>, Christopher Thomas<sup>1</sup>, Vipul Kashyap<sup>2</sup>, Amit Sheth<sup>1</sup>

<sup>1</sup> LSDIS Lab, University of Georgia,  
Athens, Georgia 30602, USA  
{cartic, cthomas, amit}@cs.uga.edu  
<sup>2</sup> LHCNCBC, National Library of Medicine,  
Bethesda, MD 20894, USA  
kashyap@nlm.nih.gov

**Abstract.** The development of taxonomies/ontologies is a human intensive process requiring prohibitively large resource commitments in terms of time and cost. In our previous work we have identified an experimentation framework for semi-automatic taxonomy/hierarchy generation from unstructured text. As observed in the preliminary results presented, the taxonomy/hierarchy quality was lower than we had anticipated. In this paper, we present two variations of our experimentation framework previously described, *viz.* Latent semantic Indexing (LSI) for document indexing and the use of term vectors to prune labels assigned to nodes in the final taxonomy/hierarchy. Using our previous results of taxonomy/hierarchy quality as the baseline we present results that demonstrate significant improvement in taxonomy/hierarchy label quality resulting from the above and present insights into the reason for the same. Finally, we present a discussion on methods for further improving taxonomy/hierarchy quality.

## 1 Introduction

Ontologies form the foundation of the Semantic Web vision [1]. A majority of the ontologies today are relatively small i.e. not of Web scale. There has been a recent trend towards the design of larger ontologies [35][34]. These efforts are however manually driven and have required prohibitively large amounts of resource commitments both in terms of time and cost. For the Semantic Web vision to be realized it is critical that along with semi-automatic approaches for semantic annotation of web resources [2][3], semi-automatic methods for ontology creation be investigated. Although there has been a recent interest in the problem of semi-automated ontology creation most of these methods have either been limited in their scope or have met with limited success. We discuss some of these in the next section. In our approach to this problem we view the creation of a taxonomy/hierarchy as the first step towards the creation of an ontology. We recognize that the taxonomy/hierarchy that we produce does not have the desirable meta-properties [33] that a “good” formal taxonomy should have. For the purposes of our work we subscribe to the following definition of

a “taxonomy/hierarchy” which is closer to that of a hierarchy or a thesaurus. “A *hierarchy or thesaurus* is a system that shows relationships between terms from general, broader concepts to more specific categories.” Furthermore, taking inspiration from information retrieval and library science, a broader term is defined as follows: a concept  $C_1$  is assumed to be *broader than* a concept  $C_2$ , if a query comprising of  $C_1$  returns a superset of the documents returned by a query comprising of  $C_2$ . We are at the first steps in our approach, where we seek to generate a hierarchy or thesaurus of concepts. We plan to consider the more formal aspects of a taxonomy as described in [33] in our future work.

In [34] we outlined the design of a comprehensive experimental framework that combines statistical clustering and NLP techniques for taxonomy/hierarchy generation. We also presented preliminary results using the SMART [32][22] indexing engine to index the document vectors in a vector space. We observed that there were many spurious labels generated for nodes in the final taxonomy/hierarchy. Further inspection revealed that these labels were terms outside the biomedical domain.

Latent Semantic Indexing (LSI) [29] has the inherent advantage over SMART in that it identifies the latent concepts or eigenvalues in the input data and maps both documents and terms into the same eigenvector space. LSI allows us to generate document and term vectors whose dimensions are the latent concepts represented by the eigenvalues. Furthermore, the selection of these underlying dimensions or latent factors involves a dimension reduction step, which chooses eigenvectors corresponding to significant eigenvalues. This potentially leads to a better and compact description of the information content represented in the underlying corpus. We believe that this feature of LSI will allow us to reduce the number of candidate labels for a node to the salient terms in the domain. We use LSI to index our dataset and also perform operations that enable us to perform a limited form of sense disambiguation on the generated taxonomy/hierarchy labels. The two main components of our approach may be summarized as:

- Use of LSI [29] to create vector-space based representations of terms and documents in a corpus based on a common set of latent dimensions
- Use of the term vectors generated above to perform sense disambiguation of the taxonomy/hierarchy labels

We compare the quality of the generated taxonomies by using NLP techniques in conjunction with SMART (as presented in our previous work) with those generated in this current work without using NLP techniques but indexing the data using an LSI based approach. This paper is organized as follows. In Section 2, we review relevant work, focusing on the attempts made by other researchers to address (parts of) this problem. The experimentation framework for taxonomy/hierarchy generation is described in detail in Section 3. In Section 4 we discuss in further detail the use of LSI for indexing. Section 5 describes in detail the techniques for node label assignment and pruning. In section 6 we briefly present the metrics we use to measure quality of the final taxonomy/hierarchy. Experiments and evaluations are presented in Section 7. Section 8 discusses the conclusions and future work.

## 2 Related Work

Approaches for semi-automatic taxonomy/hierarchy generation utilize a combination of:

- Supervised machine learning approaches that require a collection of training examples.
- NLP approaches for generating taxonomic concepts and relationships between them; and
- Clustering and data mining approaches for search, categorization and visualization of data

The concept forming system COBWEB [7] has been used to perform incremental conceptual clustering on structured instances of concepts extracted from the web [8]. An approach that used training examples consisting of structured concept instances is presented in [9]. A classification taxonomy/hierarchy based on a set of structured rules was proposed in [10]. Naïve Bayesian approaches for classification have been presented in [18].

Empirical and corpus-based NLP methods to build domain specific lexicons have been proposed in [11] and used in [12]. Approaches presented in [13] apply shallow parsing, tagging and chunking, along with statistical techniques to extract terminologies or enhance existing ontologies. In [14], a thesaurus is built by performing clustering according to a similarity measure after having retrieved triples from a parsed corpus. Linguistic structures such as verbs, appositives, nominal modifications and lexico-syntactic patterns have been used to identify is-a relationships in text [15],[16]. Salient words and phrases extracted from the documents are organized hierarchically using subsumption type co-occurrences in [17].

Effectively mining relevant information from a large volume of unstructured documents has received considerable attention in recent years [19]. Document clustering has been used for browsing large document collections in [20], using a “scatter/gather” methodology. Clustering of Web documents to organize search results has been proposed in [21]. Physicists have used clustering to find the spatial grouping of stars into galaxies [23]. An approach that pre-processes documents by applying background knowledge in order to improve the clustering results was proposed in [24]. An interesting approach proposed in this paper is that of Term Neighbourhood Expansion (TNE). This technique identifies a set of terms that are closest in the common neighbourhood of all the labels generated for a cluster. Labels generated for a cluster are typically based on an analysis of the cluster centroid.

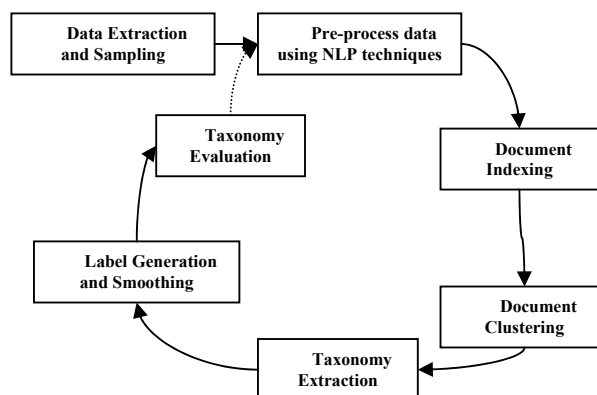
Frameworks and hybrid approaches, combining the above techniques are presented in [25][5][26]. A complementary approach that uses the structure and content of web pages on the Web to generate ontologies is presented in [27]. Hybrid approaches have also been used to automate semantic annotation, a closely related task, examples of which are the SemTag [3] and OntoMate – Annotizer systems [28], and the Semagix content management platform [4].

In our opinion our framework provides a rigorous approach to this hard problem will enable us to identify the optimal settings of various parameters that lead to semi-automated creation of useful real-world taxonomies. The novel contribution of this paper is to compare the quality of the taxonomy/hierarchy generated using Latent

Semantic Indexing (LSI) and Term Neighbourhood Expansion (TNE) in contrast with using SMART indexing in conjunction with NLP. The use of our TNE technique in conjunction with LSI considerably improves the quality of the learnt taxonomy/hierarchy.

### 3 The Taxonomy/hierarchy Generation Framework

The components of a framework for generating taxonomic/thesauri structures from textual documents are illustrated in **Figure 1**.



**Figure 1: The Taxonomy/hierarchy Generation Framework**

**Data Extraction and Sampling** MeSH and MEDLINE® are used as the target gold taxonomy/hierarchy and source of our dataset respectively. Further details are presented in Section 7.

**NLP techniques for Pre-processing** In this paper we do not pre-process the document abstracts using NLP. This is because we want to demonstrate a comparison between the LSI indexing vs. SMART indexing with the use of NLP and their effect on the taxonomy/hierarchy quality.

**Document Indexing** The abstracts (documents) are mapped to a vector space, the dimensions of which could either be words or extracted phrases. In our experiments the dimensions of the vectors in the space represent the latent dimensions generated by LSI [32]. Details of this are discussed in **Section 4**.

**Document Clustering** A bisecting K-Means strategy [30] is used to cluster our dataset. Details of our algorithm are available in [34].

**Taxonomy/hierarchy Extraction** The hierarchy generated by the above process is an artifact of the clustering process and does not capture the notion of taxonomy/hierarchy. A taxonomy/hierarchy is extracted from the cluster hierarchy using the “cohesiveness” measures. Details of this algorithm are in [34].

**Label assignment and smoothing** A set of potential labels, based on the cluster centroids are assigned to the nodes in the taxonomy/hierarchy. Various techniques such as propagation of labels to parent nodes and *TNE (Term Neighborhood Expansion)* is used to prune labels of node in the final taxonomy/hierarchy.

**Taxonomy/hierarchy Evaluation** Finally, the generated taxonomy/hierarchy is evaluated *wrt* the gold standard taxonomy/hierarchy using a variety of different measures that measure content-based similarity (i.e., overlap between the labels extracted) and the structural similarity (i.e., consistency of parent-child relationships) between the two taxonomies. Section 6 explains our metrics in some detail.

Definitions of some symbols used are in order. Consider a set of document vectors  $\mathbf{D} = \{d_1, \dots, d_M\}$  in the Euclidean space  $\mathbf{R}^N$ . Let the centroid of the set be denoted by:

$$m(\mathbf{D}) = \frac{1}{M} \sum_{i=1}^M d_i$$

The intra-cluster (or intra-set) cohesiveness is defined as:

$$c(\mathbf{D}) = \frac{1}{M} \sum_{i=1}^M \cos(d_i, m(\mathbf{D}))$$

Let  $\{\pi_i\}_{i=1}^k$  be a partition of  $\mathbf{D}$  with the corresponding centroids  $m_1 = m(\pi_1), \dots, m_k = m(\pi_k)$ . The parent/child relationships are established as follows:

$$\forall i, 1 \leq i \leq k, \text{child}(\mathbf{D}) = \pi_i \text{ and } \text{parent}(\pi_i) = \mathbf{D}$$

The quality of the partition increases if the intra-cluster cohesiveness increases. Thus the quality  $\mathcal{Q}$  of the partition  $\{\pi_i\}_{i=1}^k$  is given by:

$$\mathcal{Q}(\{\pi_i\}_{i=1}^k) = \frac{1}{k} \sum_{i=1}^k c(\pi_i)$$

Given a cluster node  $\pi_i$ , we define the *labels*( $\pi_i$ ) to contain the labels extracted from the cluster centroid.

The objective in using NLP pre-processing using a chunk parser in our earlier work was to identify the features that contributed most to the meaning of a document. We therefore extracted noun-phrases from our datasets and used these as features of document vectors. As discussed earlier, LSI allows us to extract the salient concepts in the domain and generates document and term vectors whose dimensions are the latent concepts represented by the eigenvalues. Our objective in this paper is therefore to compare the above methods for feature extraction in the context of semi-automated taxonomy/hierarchy generation.

## 4 Document indexing using Latent Semantic Indexing

Using SVD analysis to generate and LSI vector space LSI applies singular-value decomposition (SVD) to a term-document matrix where each entry gives the number of times a term appears in a document [37]. Typically, a large term-document matrix is decomposed to a set of 200-300 orthogonal factors from which the original matrix can be approximated by linear combination. Roughly speaking, these factors may be thought of as artificial concepts; they represent extracted common meaning components of many different words and documents. Each term or document is then characterized by a vector of weights indicating its strength of association with each of these underlying concepts.

Consider a collection of  $m$  documents with  $n$  unique terms that, together, form an  $n$  by  $m$  sparse matrix  $E$  with terms as its rows and the documents as its columns. Each entry in  $E$  gives the number of times a term appears in a document. In the usual case,

log-entropy weighting ( $\log(tf+1)$ entropy) is applied to these raw frequency counts before applying SVD. The structure attributed to document-document and term-term dependencies is expressed mathematically in the SVD of E:

$$E = U(E) \Sigma(E) V(E)^T$$

where  $U(E)$  is an  $n \times n$  matrix such that  $U(E)^T U(E) = I_n$ ,  $\Sigma(E)$  is an  $n \times n$  matrix of singular values and  $V(E)$  is an  $n \times m$  matrix such that  $V(E)^T V(E) = I_m$ , assuming for simplicity that  $E$  has fewer terms than documents. The attraction of SVD is that it can be used to decompose  $E$  to a lower dimensional vector space  $k$ . In this rank-k construction:

$$E = U_k(E) \Sigma_k(E) V_k(E)^T$$

In this LSI vector space, words similar in meaning and documents with similar content will be located near one another. These dependencies enable one to query documents with terms, but also terms with documents, terms with terms, and documents with other documents. Berry, Dumais and O'Brien [37] provide a formal justification for using the matrix of left singular vectors  $U_k(E)$  as a vector lexicon.

The use of LSI allows us to apply novel technique to prune labels as discussed in Section 5. This technique is referred to as *Term Neighborhood Expansion (TNE)*. Let  $labels(\pi_i)$  represent the labels in a node. Let  $l_j \in labels(\pi_i)$ . Further, let us define  $neighborhood(l_j)$  as the set of labels that are "closest" to the term  $l_j$ .

$$neighborhood(l_j) = \{t \mid t \in \text{Max}_k(\{\vec{t}_i \cdot \vec{l}_j\}), t_i \in \text{term vector lexicon}\}$$

$\text{Max}_k(S)$  denotes the top  $K$  elements of a set. Here "closest" is determined by computing the cosine of each term vector in the corpus with the vector corresponding to  $l_j$  and choosing the top  $k$ . Therefore,

$$N(\pi_i) = \bigcup_{1 \leq j \leq n} neighborhood(l_j) \text{ where } l_j \in labels(\pi_i),$$

and  $n$  is the number of labels in node  $\pi_i$ . Let  $l_m \in N(\pi_i)$  and  $W(l_m)$  represent the weight of label  $l_m$ . Thus the weight of each label  $l_m \in N(\pi_i)$  is computed as follows

$$W(l_m) = \sum_{1 \leq j \leq n} w(l_m), \text{ where } l_m \in neighborhood(l_j) \wedge w(l_m) = |\vec{l}_m \bullet \vec{l}_j|$$

It should be noted that  $W(l_m) = 0$ , if  $l_m \notin neighborhood(l_j) \quad \forall j \mid 1 \leq j \leq n$ .

Once the weight of each of the labels in  $N(\pi_i)$  is determined, the top  $k$  terms from these are chosen as the labels for this node.

## 5 Taxonomy/hierarchy Node Labeling

There are essentially two aspects of node labeling:

- Label propagation and smoothing using the label propagation algorithm discussed below

- Determining the dominant sense neighborhood using Term Neighborhood Expansion (TNE)

In [34] we assigned labels to each node, based on the node’s centroid vector. Since we were using SMART [32], we simply chose the top  $k$  weighted values of the centroid vector and determined the salient terms. In this paper the use of LSI [29] means that terms and documents are represented in the same “latent” space. This enables us to compute the (cosine) distance between the centroid vector and the term vectors and thereby find the terms that are “closest” to the centroid. In our experiments in this paper we have used this technique to generate candidate labels for nodes. Given a cluster node  $\pi_i$ , we define the  $labels(\pi_i)$  to contain the labels extracted from the cluster centroid.

$$\begin{aligned} \text{childLabels}(\pi) &= \bigcup_{A \in \text{children}(\pi)} \text{labels}(A) \\ \text{parentLabels}(\pi) &= \text{labels}(\text{parent}(\pi)) \\ \text{taxLabels}(\mathcal{T}) &= \bigcup_{A \in \mathcal{T}} \text{labels}(A) \end{aligned}$$

The same labels can appear in multiple nodes of the taxonomy/hierarchy. Our approach for refining the labels of the taxonomy/hierarchy, referred to as *taxonomic propagation*, involves propagation of labels across different levels of the taxonomy/hierarchy. Some heuristics used are:

- **Propagate to Child:** If a label appears both in the parent and one or few children, the label will be propagated to the child and removed from the parent. A parent node in a taxonomy/hierarchy is a generalization of its children. Hence the parent should not have a label that only one or few of its children have.
- **Propagate to Parent:** If a label has been assigned to all the children of a node, the label will be propagated to the parent and removed from all the children nodes at which it appears. If every child of a node in a taxonomy/hierarchy has a label that the node itself has, having that label in the parent node suffices to convey the fact that children of this node also talk about the concept that the label represents.

The algorithm for label propagation is as follows:

1. Start with the Root( $\mathcal{T}$ )
2. For each cluster node  $\pi_i$  at level L do
  - a. For cluster node  $\pi_j \in \text{children}(\pi_i)$  do
    - i. If  $\Delta = \text{labels}(\pi_i) \cap \text{labels}(\pi_j) \neq \phi$
    - ii.  $\text{labels}(\pi_i) = \text{labels}(\pi_i) - \Delta$
3. End Propagate to Children
4. Start with cluster nodes in leaves( $\mathcal{T}$ )
5. For each cluster node  $\pi_i$  at level L do
  - a. If  $\Delta = \text{labels}(\pi_i) \cap \text{childLabels}(\pi_i) \neq \phi$
  - b.  $\text{labels}(\pi_i) = \text{labels}(\pi_i) + \Delta$
  - c. For  $\pi_j \in \text{children}(\pi_i)$  do
    - i.  $\text{labels}(\pi_j) = \text{labels}(\pi_j) - \Delta$
6. End Propagate to Parent
7. End Label Propagation

After the label propagation stage we apply *Term Neighborhood Expansion (TNE)*, discussed in the previous section, which attempts to further reduce the number of

potential labels for each node in the final taxonomy/hierarchy. This technique computes the neighborhood of each label as discussed before,

## 6 Taxonomy/hierarchy Quality Metrics

We separate the content and structural aspects of a taxonomy/hierarchy, in an attempt to discover trade-offs and dependencies that might exist between the two. A discussion of the various quality metrics is presented below:

- **Content Quality Metric (CQM):** This measures the overlap in the labels present in the generated Taxonomy/hierarchy,  $T_{gen}$  and the gold standard taxonomy/hierarchy (subtree of MeSH rooted at “Neoplasms”)  $T_{gold}$ . There are two variants of this metric:

- **CQM-P:** This measures the precision, i.e., the percentage of labels in  $T_{gen}$  that appear in  $T_{gold}$

$$CQM-P = \frac{|\text{taxLabels}(T_{gen}) \cap \text{taxLabels}(T_{gold})|}{|\text{taxLabels}(T_{gen})|}$$

- **CQM-R:** This measures the recall, i.e., the percentage of labels in  $T_{gold}$  that appear in  $T_{gen}$

$$CQM-R = \frac{|\text{taxLabels}(T_{gen}) \cap \text{taxLabels}(T_{gold})|}{|\text{taxLabels}(T_{gold})|}$$

- **Structural Quality Metric (SQM):** This measures the structural validity of the labels, i.e., when two labels appear in a parent child relationship in  $T_{gold}$ , they should appear in a consistent relationship (parent-child or ancestor-descendant) in  $T_{gen}$  or vice versa. Based on the above discussion, let:

$$pcLinks(T) = \{ \langle a, b \rangle \mid a \text{ is parent of } b \text{ in } T \}$$

$$adLinks(T) = \{ \langle a, b \rangle \mid a \text{ is ancestor of } b \text{ in } T \}$$

$$adLinks(T) \supseteq pcLinks(T)$$

As above, there are two variants of the SQM metric:

- **SQM-P:** This measures the precision, i.e., the percentage of parent-child relationships in  $T_{gen}$  that appear consistently in  $T_{gold}$ .

$$SQM-P = \frac{|\text{pcLinks}(T_{gen}) \cap \text{adLinks}(T_{gold})|}{|\text{pcLinks}(T_{gen})|}$$

- **SQM-R:** This measures the recall, i.e., the percentage of parent-child relationships in  $T_{gold}$  that appear consistently in  $T_{gen}$ .

$$SQM-R = \frac{|\text{pcLinks}(T_{gold}) \cap \text{adLinks}(T_{gen})|}{|\text{pcLinks}(T_{gold})|}$$

The above measures are scaled appropriately. It is quite likely, especially for smaller data sets, that the number of concepts generated in  $T_{gen}$  is likely to be less than the number of concepts in  $T_{gold}$ . This will have an impact on the recall related

quality measures and the respective denominators in CQM-R and SQM-R will be scaled appropriately to reflect this.

## 7 Results and Observations

A *gold standard taxonomy/hierarchy* (MeSH [6]) is chosen and relevant citations are sampled from the MEDLINE® bibliographic database. We chose the sub-tree under the concept *Neoplasms* consisting of 649 concepts. Multiple data sets of different sizes are sampled using techniques such as uniform or density biased sampling, based on the underlying distribution of the documents *wrt* the concepts in the taxonomy/hierarchy. Each data point in the graphs shown below is obtained by averaging the values of quality generated by 10 sample datasets. These samples are created using density biased sampling [31]. The results that we present in this paper are compared to the baseline results that we obtained in our earlier work [34]. We recognize that MeSH does not have some of the desirable meta-properties [33] that a “good” taxonomy/hierarchy should have. Although the word “taxonomy/hierarchy” is loosely used to describe MeSH, we do not subscribe the view that MeSH is a taxonomy/hierarchy from the formal viewpoint and is a hierarchy of concepts or thesaurus<sup>1</sup> at best. MeSH is however one of the most widely and effectively used organizations of concepts in the biomedical field. It has been created by domain experts and is used to index over 14 million medical documents. These features have been the majors deciding factor in our choice of MeSH and MEDLINE® as our data sources.

The notion of differentiation is captured by the difference in the *cluster cohesiveness* between successive layers of the hierarchical cluster tree. In the course of our experimentation, it was observed that the successive values of cohesiveness down a cluster hierarchy are *monotonically increasing* in value. Details of this algorithm are discussed in [34]. The taxonomy creator or user is expected to suggest a set of cohesiveness levels which correspond to differentiation between the various layers of the taxonomy. This is the extent of human involvement in the overall process. It is however possible to determine the optimal setting of these cohesiveness levels that maximize the overall quality of the output taxonomy. In fact we plan to use a Genetic Algorithm [38] to determine the maximum quality measures for a particular configuration of our framework. As it turns out this is a multi-objective optimization problem [38]. For the purposes of this paper we have simply chosen to run the taxonomy/hierarchy extraction with *no. of cohesiveness levels* = 6,8,10 and 15 values of cohesiveness where the cohesiveness values are computed by dividing the range of values observed into  $n$  parts and choosing the boundaries that divide these parts as the cohesiveness values. Among the runs of the taxonomy/hierarchy extraction process using the above values the one that gave us the maximum value for the label quality recall was chosen. These results are shown below. Figure 2 shows the values of label recall CQM-R obtained using LSI in conjunction with TNE in contrast with those obtained with SMART using NLP. As is evident from the figure the improvement obtained by using LSI with TNE is appreciable. Figure 3 shows the same comparison

---

for the structure precision SQM-P. A similar trend is observed in this comparison too. The values obtained using LSI+TNE are better than those obtained using SMART and NLP preprocessing. Figure 4 shows the comparison between structural recall between the two methods. Clearly the use of LSI with TNE results in an overall increase in the quality of the taxonomy produced. Another observation is that the use of LSI and TNE makes the quality of the final taxonomy independent of the number of labels extracted. This is evident from the smoothing effect achieved in the graphs for LSI and TNE. This smoothing effect was observed even when larger values of  $k$  (number of labels extracted) were used.

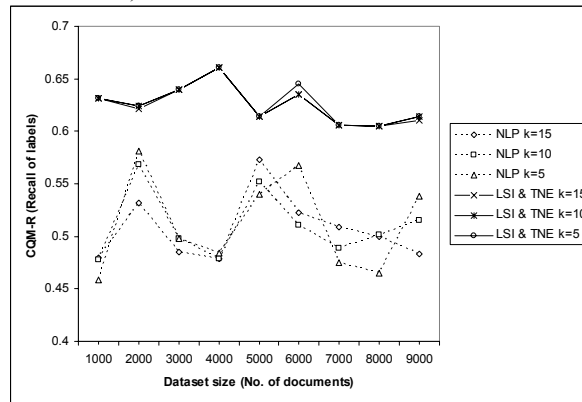


Figure 2 Comparison of Content Quality Recall (LSI+TNE vs. SMART+NLP)

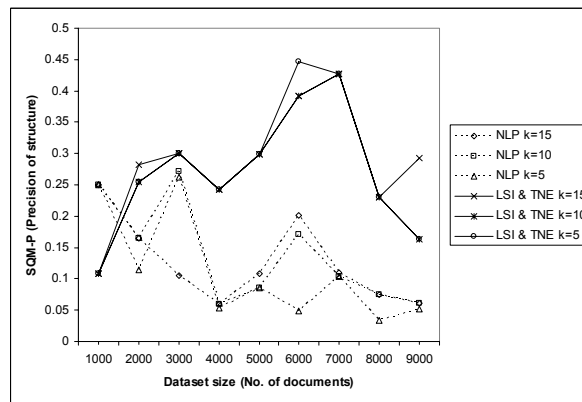
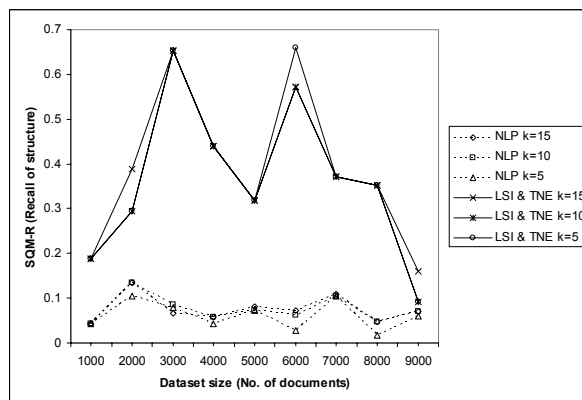


Figure 3 Comparison of Structure Quality Precision (LSI+TNE vs. SMART+NLP)



**Figure 4 Comparison of Structure Quality Recall (LSI+TNE vs. SMART+NLP)**

The use of LSI to index a documents and vectors into the same eigenvector space allows us to reduce the number of dimensions along which both term vectors and document vectors differ. These dimensions indirectly represent the latent salient “concepts” in the corpus. In addition to this our TNE technique begins with a set of labels assigned to a node and further reduces it by finding the dominant set of cohesive terms for that node. It does this by using the term vector lexicon generated by LSI to compute a restricted set of labels for each node in the taxonomy/hierarchy. These techniques together reduce the number of labels of nodes in a taxonomy while ensuring that these labels are the salient domain terms. This is the reason why our results show an increase in overall quality.

## 8 Conclusions and Future Work

In this paper we built upon our previous work by investigating the use of LSI in conjunction with TNE to improve the quality of the taxonomy/hierarchy generated. Our results have shown that these techniques produce a substantial increase in label recall (CQM-R), structure recall (SQM-R) and structure precision (SQM-P) over our baseline (results from SMART with NLP). These results seem to obviate the use of shallow NLP preprocessing. This however does not necessarily rule out the use of stronger NLP techniques to improve the taxonomy/hierarchy quality further. We therefore propose to use stronger NLP techniques at various stages of our framework to further improve our results. We also propose to investigate the used of hyponymy, hypernymy and synonymy among other relations in WordNet® to further reduce the number of labels assigned to the nodes. Our results also pointed to the possibility of running a multi-objective optimization to determine the optimal values of content and structure recall/precision measure based on different values of the cohesiveness used to extract a taxonomy/hierarchy. This will provide a maximal limit of quality measure

against which other variations can be measured. The performance improvement that we have demonstrated in this paper provides an initial validation of our overall framework, approach and techniques. We believe that we have made substantial progress toward the goal of semi-automatic taxonomy/hierarchy generation.

## Acknowledgements

We would like to thank Hui Han, Hongyuan Zha from Penn State University and John Wilbur from NCBI (National Center for Biotechnology Information) for providing us with access to LSI and for stimulating discussions on the topic.

## References

- [1] T. Berners Lee, J. Hendler and O. Lassila. The Semantic Web. *Scientific American*, May 2001.
- [2] S. Handschuh, S. Staab and R. Volz. On Deep Annotation. *Proceedings of the 12<sup>th</sup> International WWW Conference (WWW 2003)*, Budapest, Hungary, May 2003.
- [3] S. Dill et. al. SemTag and SemSeeker: Bootstrapping the Semantic Web via automated semantic annotation. *Proceedings of the 12<sup>th</sup> International WWW Conference (WWW 2003)*, Budapest, Hungary, May 2003.
- [4] B. Hammond, A. Sheth, and K. Kochut. Semantic Enhancement Engine: A Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content. In *Real World Semantic Web Applications*, V. Kashyap and L. Shklar, Eds., IOS Press.
- [5] C Y Chung, R. Lieu, J. Liu, A. Luk, J. Mao and P. Raghavan. Thematic Mapping – From Unstructured Documents to Taxonomies. *Proceedings of the 11<sup>th</sup> International Conference on Information and Knowledge Management (CIKM 2002)*, McLean, VA, November 2002.
- [6] MeSH. Medical Subject Headings. Bethesda (MD): National Library of Medicine, 2003. <http://www.nlm.nih.gov/mesh/meshhome.html>
- [7] D. H. Fisher. Knowledge Acquisition via incremental conceptual clustering. *Machine Learning 2:139-172*, 1987
- [8] P. Clerkin, P. Cunningham and C. Hayes. Ontology Discovery for the Semantic Web using Hierarchical Clustering. *Proceedings of the Semantic Web Mining Workshop co-located with ECML/PKDD 2001*, Freiburg, Germany, September 2001
- [9] W. W. Cohen and H. Hirsh. Learning the CLASSIC Description Logic: Theoretical and Experimental Results. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourth International Conference*, 1994.
- [10] H. Suryanto and P. Compton: Learning Classification taxonomies from a classification knowledge based system. In *Proceedings of Workshop on Ontology Learning at ECAI-2000*, 2000.

- [11] E. Riloff and J. Shepherd. A corpus-based approach for building semantic lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, Providence, RI, 1997.
- [12] S. Dill et. al. SemTag and SemSeeker: Bootstrapping the Semantic Web via automated semantic annotation. *Proceedings of the 12<sup>th</sup> International WWW Conference (WWW 2003)*, Budapest, Hungary, May 2003.
- [13] B. Daille. Study and implementation of combined techniques for automatic extraction of terminology. In P. Resnick and J. Klavans, editors, *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, MIT Press, Cambridge, MA, 1996.
- [14] D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL-98*, 1998.
- [15] Hearst, M.: *Automatic acquisition of hyponyms from large text corpora*. In Proceedings of the 14th International Conference on Computational Linguistics. Nantes, France, 1992.
- [16] M. Fiszman, T. C. Rindflesch and H. Kilicoglu. Integrating a Hypernymic Preposition Interpreter into a Semantic Processor for Biomedical Texts. In *Proceedings of the AMIA Annual Symposium on Medical Informatics*, 2003.
- [17] M. Sanderson and B. Croft. Deriving Concept Hierarchies from Text. International Conference on Research and Development in Information Retrieval (SIGIR 1999), 1999.
- [18] S. Chakrabarti, B. Dom, R. Agrawal and P. Raghavan. Using taxonomy/hierarchy, discriminants, and signatures to navigate in text databases. In *VLDB*, Athens, Greece, September 1997.
- [19] S. Chakrabarti. Data Mining for Hypertext: A Tutorial Survey. *ACM SIGKDD Explorations*, 1(2):1-11, 2000.
- [20] D. R. Cutting, D. R. Karger, J. O. Pedersen and J. W. Tukey. Scatter/Gather: A cluster-based approach to browsing large document collections. In *Annual International Conference on Research and Development on Information Retrieval*, Denmark, 1992.
- [21] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. In *Proceedings of ACM SIGIR Conference*, 1998.
- [22] C. Buckley, M. Mitra, J. Walz and C. Cardie. Using clustering and superconcepts within SMART: TREC 6. In *Sixth Text Retrieval Conference (TREC-6)*, Gaithersburg, MD, November 1997.
- [23] J. Kepner, X. Fan, N. Buhcall, J. Gunn, R. Lupton and G. Xu. An Automated Cluster Finder: The Adaptive Matched Filter. *The Astrophysics Journal*, 517, 1999.
- [24] A. Hotho, S. Staab and A. Maedche. Ontology-based Text Clustering. In *Proceedings of the IJCAI 2001 Workshop on Text Learning: Beyond Supervision*, Seattle, USA, 2001.
- [25] A. Maedche and S. Staab. Ontology learning for the Semantic Web. *IEEE Intelligent Systems*, 16, 2001.
- [26] M. Reinberger, P. Spyns, W. Daelemans and R. Meersman. Mining for Lexons: Applying unsupervised learning methods to create ontology bases. In *Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics (ODBASE)*, November 2003.
- [27] H. Davulcu, S. Vadrevu and S. Nagarajan. OntoMiner: Bootstrapping and Populating Ontologies from Domain Specific Websites. *Proceedings of the First International Workshop on Semantic Web and Databases (SWDB 2003)*, Berlin, September 2003.

- [28] S. Handschuh, S. Staab and R. Volz. On Deep Annotation. *Proceedings of the 12<sup>th</sup> International WWW Conference (WWW 2003)*, Budapest, Hungary. May 2003.
- [29] Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A., "Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6), 391-407, 1990.
- [30] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.
- [31] C. R. Palmer and C. Faloutsos. Density Biased Sampling: An Improved Method for Data Mining and Clustering. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, May 2000
- [32] G. Salton. *The SMART Retrieval System – Experiments in Automatic Document Retrieval*, Prentice Hall, 1971.
- [33] Nicola Guarino and Christopher Welty. *A Formal Ontology of Properties*. In *Proceedings of the ECAI-00 Workshop on Applications of Ontologies and Problem Solving Methods*, pp. 12-1 12-8, Berlin, Germany, 2000a.
- [34] TaxaMiner: An Experimentation Framework for Automated Taxonomy/hierarchy Bootstrapping. V Kashyap, C. Ramakrishnan, C. Thomas, D. Bassu, T. C. Rindfleisch and A. Sheth Technical Report, Computer Science Dept., University of Georgia, March 5<sup>th</sup> 2004 <http://lstdis.cs.uga.edu/~cthomas/resources/taxaminer.pdf>
- [35] Aleman-Meza, B., Halaschek, C., Sheth, A., Arpinar, I.B., Sannapareddy, G.: SWETO: Large- Scale Semantic Web Test-bed. *Intl. Workshop on Ontology in Action*, Banff, Canada, June 20-24, 2004
- [36] Guha, R., McCool, R.: TAP: A Semantic Web Test-bed. *Journal of Web Semantics*, Volume 1, Issue 1, December (2003)
- [37] Berry, M.; Dumais, S.; and O'Brien, G. 1995. Using Linear Algebra for Intelligent Information Retrieval. *SIAM Review*, 35(4).
- [38] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Co., Inc., 1989

## Appendix

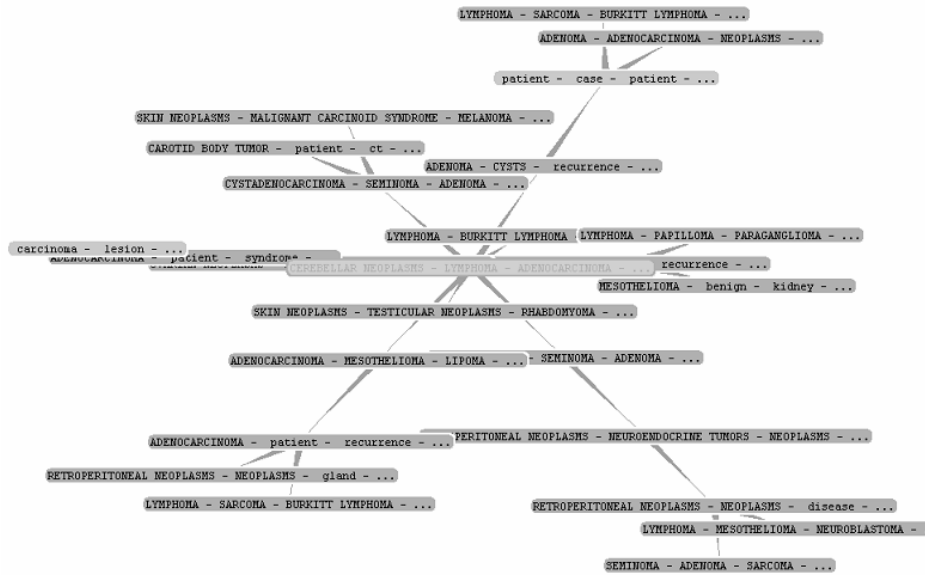


Figure 5 Example of Learnt Taxonomy/Hierarchy (note that the darker shaded nodes and capitalized labels indicate a match with a gold taxonomy node)

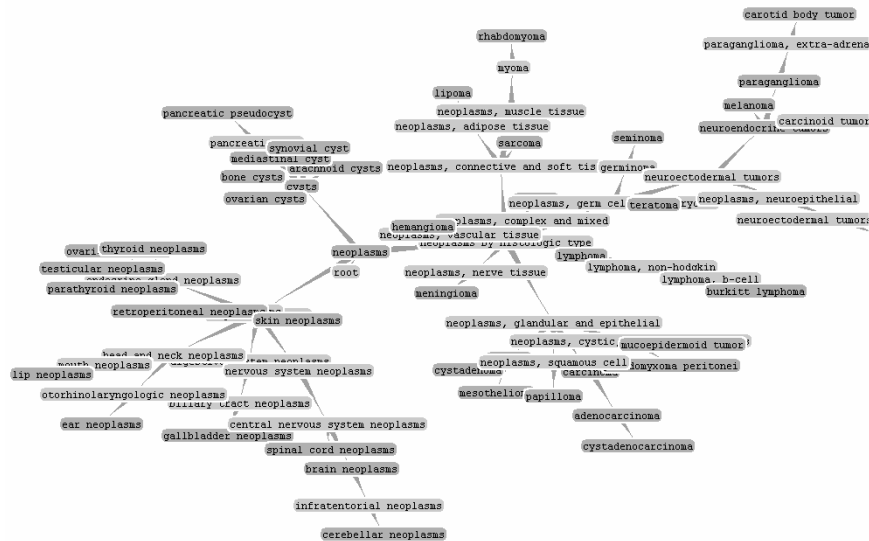


Figure 6 Corresponding portion of MeSH (gold taxonomy/hierarchy)