2005

TaxaMiner: An Experimentation Framework for Automated Taxonomy Bootstrapping

Vipul Kashyap

Cartic Ramakrishnan
Wright State University - Main Campus

Christopher Thomas

Amit P. Sheth
Wright State University - Main Campus, amit@sc.edu

TaxaMiner: An Experimentation Framework for Automated Taxonomy Bootstrapping

V Kashyap[1], C Ramakrishnan[2], C Thomas[2] and A Sheth[2]

[1]Clinical Informatics R&D, Partners HealthCare System, 93 Worcester St, Wellesley, MA 02481
[2]LSDIS Lab, Department of CS, University of Georgia, 415 GSRC, Athens, GA 30602
vkashyap1@partners.org

ABSTRACT

Ontologies are a central component of the Semantic Web (SW) infrastructure. The design and construction of domain ontologies and taxonomies is a human intensive process which requires allocation of huge resources in terms of cost and time. For the SW to scale and become feasible, approaches that reduce human effort and resource commitments need to be investigated urgently. Towards this end, we present a framework for automated taxonomy construction based on a large corpus of documents, a first step towards large scale, automated ontology construction. Our approach involves: (a) generation of a document cluster hierarchy; (b) extraction of a topic hierarchy from this cluster hierarchy; and (c) assignment of labels to nodes in the topic hierarchy. We draw upon a suite of clustering and NLP techniques and identify parameters which form the basis of an experimentation framework. We also propose metrics to measure quality of the resulting topic hierarchy and evaluate the impact of various parameters on these quality metrics. The MEDLINE® database is used as the document corpus and the MeSH thesaurus as the gold standard. Insights from these experiments are presented and discussed.

1. INTRODUCTION

The Semantic Web (SW) [1] has been proposed as an extension to the current Web where the content will be machine-understandable. This content is likely to be in the form of documents annotated with metadata descriptions, or data stored in back-end relational databases mapped to structured ontologies (or schemata) describing content in a domain specific manner. Software programs or agents will then be able to gather and analyze information over the web, enabling the development of software to assist humans and streamline business processes both within and across organizational boundaries.

However, machines today understand very little of available web content. In fact, most of the annotations are in the form of tags that describe structure, formatting or presentation information. Approaches for annotation have primarily been manual [2][3], though there have been some attempts at exploring semi-automatic approaches for metadata annotation [4][48]. As observed in these efforts, two resources necessary for realizing the semantic web are: (a) large scale availability of domain specific ontologies; and (b) large scale availability of annotations or metadata descriptions created by using terms, concepts or relationships provided by these ontologies. In this paper, we focus on the former, i.e., addressing the need for domain specific ontologies.

Ontologies are a central component of the SW infrastructure. However, it is well acknowledged that design and construction of ontologies is a labor-intensive process and requires allocation of huge resources in terms of cost and time. For the SW vision to be realized and scale up, it is critical to investigate approaches that reduce human effort and resource commitments. Whereas, the broad goal of the endeavor should be semi-automatic creation of domain ontologies, we begin with an attempt to create an initial thesaurus/taxonomy of concepts using a largely unsupervised learning approach. This taxonomy forms the vital first step in bootstrapping ontologies from textual documents that form an overwhelming proportion of content available on the Web today.

Previous work has investigated desirable properties of a "good" taxonomy [49] and generation of topic hierarchies from text has been investigated in [27][50]. For the purposes of our work we subscribe to the following definition: *"A **taxonomy** is a system of knowledge organization that represents relationships between topics such that they arrange these concepts from general, broader concepts to more specific concepts."* We define a broader concept as follows: ***a concept C_1 is said 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***. In the first steps in our approach, we seek to generate a taxonomy of concepts. Unlike [50], however, we aim to extract taxonomies that are substantially broader and deeper than those required to summarize search results and describe the knowledge domain in some manner. We plan to consider the more formal aspects of a taxonomy as described in [49] in our future work.

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 generation is described in detail in Section 3. The various components of the framework are discussed in detail in Sections 4-9. In Section 10 we present metrics for measuring taxonomy quality in the context of experiments and evaluations. Section 11 discusses the conclusions and future work.

2. RELATED WORK

Approaches for semi-automatic generation of ontologies or taxonomies from underlying content may be characterized as:

- Supervised machine learning based approaches, which require a large number of training examples, traditionally generated manually.
- NLP approaches applied for generating ontological concepts and relationships. These are based on rules that analyze patterns based on syntactic categories, which requires significant human involvement, making it expensive and infeasible for large scale SW applications.
- Statistical Clustering methods have been used to partition data sets, categorize search results and visualize data. However, they have not focused on generating labels for clusters and creation of new taxonomies.

Machine learning approaches are for the most part supervised, where a set of manually generated positive and negative training examples are used. An approach using the concept forming system COBWEB [16] has been used to perform incremental conceptual clustering on structured instances of concepts extracted from the web [10]. Experimental and theoretical results on learning the CLASSIC description logic were presented in [32], and were used to construct concept hierarchies. An approach to bootstrap a classification taxonomy based on a set of structured rules was proposed in [35]. A supervised approach presented in [34], supports semi-automatic and incremental bootstrapping of a domain-specific information extraction system.

Empirical and corpus-based NLP methods to build domain specific lexicons have been proposed in [11] and used in [4]. Approaches that learn meanings of unknown words based on other word definitions in the surrounding context have been presented in [12][13]. Case-based methods, that match unknown word contexts against previously seen word contexts are described in [14][15]. Approaches presented in [25][26] apply shallow parsing, tagging and chunking, along with statistical techniques to extract terminologies or enhance existing ontologies. Full parse tree construction followed by decomposition into elementary dependency trees has been used to create medical ontologies from French text corpora in [29]. In [30], 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 and nominal modifications have been used to identify hypernymic propositions in the biomedical text [17]. Lexico-syntactic patterns have been investigated for inferring hyponymy from textual data in [7]. Salient words and phrases extracted from the documents are organized hierarchically using subsumption type co-occurrences in [27]. A description of supervised and unsupervised approaches to extract semantic relationships between terms in a text document is presented in [24]. A generalized association rule algorithm proposed in [31] detects non-taxonomic relationships between concepts and also determines the right level of abstraction at which to establish the relationship.

Effectively mining relevant information from a large volume of unstructured documents has received considerable attention in recent years [18][19][20]. A survey on the use of clustering in Information Retrieval is presented in [40]. Document clustering has been used for browsing large document collections in [21], using a "scatter/gather" methodology. These approaches create vector space representations of documents and use Euclidean or cosine distance-based similarity metrics like the Euclidean to extract clusters from groups of documents. Clustering of Web documents to organize search results has been proposed in [22][38]. Physicists have used clustering to find the spatial grouping of stars into galaxies [39]. An approach that pre-processes documents by applying background knowledge in order to improve the clustering results was proposed in [23].

An interesting framework for hybrid approaches, combining the above techniques is presented in [36]. The Thematic Mapping System [8] developed at Verity, Inc. and the lexon mining approach [28] most closely reflects our perspective. A complementary approach that uses the structure and content of HTML-based pages on the Web to generate ontologies is presented in [9]. Hybrid approaches have also been used to automate semantic annotation, a closely related task, examples of which are the SemTag [4] and OntoMate – Annotizer systems [3], and the Semagix content management platform [48].

In view of the above interesting work based on component technologies, we present a comprehensive framework that combines some of these components, and consists of the following novel features:

- An experimental framework combining Statistical Clustering, NLP and other customized techniques for taxonomy generation.
- Exploitation of the statistics generated during the clustering process to extract a more meaningful taxonomy. Identification of statistical parameters that characterize the notion of "differentiation" in the taxonomic structure.
- Techniques for automatic generation and refinement of labels for nodes in the final taxonomy.
- Investigation of the impact of various components of the framework on the quality of the taxonomy generated, based on metrics designed for this purpose.
- Initial validation of our approach using a real world data set, the MEDLINE® database and real world taxonomy, the MeSH thesaurus.

3. THE TAXONOMY GENERATION FRAMEWORK

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

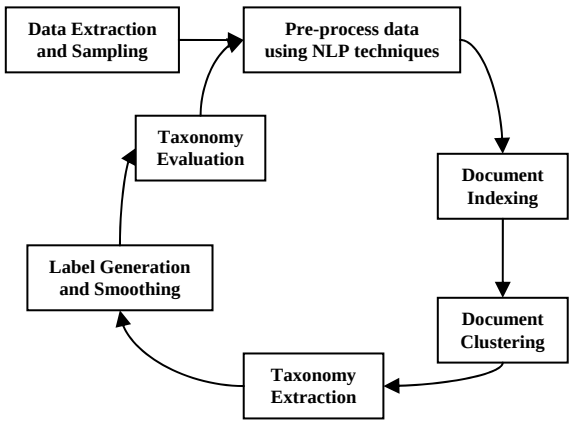

Figure 1: The Taxonomy Generation Framework

Data Extraction and Sampling MeSH and MEDLINE® are used as the gold standard hierarchy and source of our dataset respectively. We use density-biased sampling [37] to sample documents from MEDLINE for our experiments. Details of this are discussed in Section 4.

NLP techniques for Pre-processing NLP techniques such as Part-of-Speech Tagging and Chunk Parsing are used to extract noun phrases from the citation abstracts. These phrases may be simple (1-2 words long), macro (2-3 words long) or mega (3-5 words long). Details of this are discussed in Section 5. Another option is to use LSI to index documents without any NLP pre-processing. We compare these two options in our results.

Document Indexing The document abstracts are mapped to a vector space, the dimensions of which could either be words or

extracted phrases. In our experiments we use both SMART [6][38] and LSI [47]. We compare the performance of the both indexing methods in terms of the quality of the resulting Topic Hierarchy. Details of this are discussed in Section 6**.**

Clustering the dataset. A bisecting K-Means **Error! Reference source not found.** strategy or a Principal Direction Divisive Partitioning approach [51] may be used to cluster our dataset. One could perform term-based clustering vs document-based clustering approaches We use a variant of the K-means algorithm for document clustering, details of which are provided in Section 7.

Taxonomy Extraction The hierarchy generated by the above process is an artifact of the clustering process. It is at best a "history" of the clustering process. However cluster cohesiveness measures are computed for each cluster. A taxonomy is extracted from the cluster hierarchy using these cohesiveness measures as a guide. Details of this algorithm are in Section 8.

Label assignment and smoothing A set of potential labels, based on the cluster centroids are assigned to the nodes in the extracted taxonomy. Various techniques such as propagation of labels to parent nodes and *TNE (Term Neighborhood Expansion)* are used to refine labels in the final taxonomy. Details of this are presented in Section 9.

Taxonomy Quality Evaluation Finally, the generated taxonomy is evaluated *wrt.* the gold standard using a variety of different metrics 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 hierarchies. Section 10 explains our metrics in some detail.

We now discuss the individual components of the Taxonomy Generation Framework in greater detail.

4. SAMPLING THE DATA SET

A subset of the MEDLINE® bibliographic database satisfying the following conditions is extracted: (a) the MEDLINE® citation should be annotated by one of the 649 concepts present in the gold taxonomy, i.e. the MeSH sub-tree under the concept *Neoplasms*; (b) the concepts that annotate the citation should be identified as "preferred"; and (c) the citation should have a non-empty abstract.

MeSH, which is used as the gold standard in our experiments, while not a taxonomy in the formal sense from a knowledge representation viewpoint, is however on the most widely used organizations of concepts in the biomedical field. It has been created by domain experts and is used to index over 14 million MEDLINE® citations. These features have influenced us in our choice of the MeSH as the gold standard taxonomy and the MEDLINE® as the experimental data set.

"Uniform random sampling is frequently used in practice and also frequently criticized because it will miss small clusters. Many natural phenomena are known to follow Zipf's distribution and the inability of uniform sampling to find small clusters is of practical concern" [37]. In the context of our approach, sampling is likely to be biased in such a way as to produce a taxonomy containing concepts which appear only in a large number of MEDLINE® citations. Hence, we adopt the approach of density biased sampling as proposed in [37] where we probabilistically under-sample dense regions, i.e., concepts that appear as annotations of a large number of MEDLINE® citations; and over-sample light regions, i.e., concepts that appear as annotations of a small number of MEDLINE® citations. Density biased sampling relies on the *a priori* approximate grouping of data points in the sample. It then samples points from these groups whilst ensuring that dense regions are under-sampled and sparse regions over-sampled. The advantage we have in our experiment is that we know exactly what these groups are *a priori*. This enables us to greatly simplify the sampling process in our experiments. As discussed in [37], the data sets sampled have the following characteristics:

- Given a MeSH concept, documents are selected with a uniform probability. The probability function is:

$$\mathrm{f}(\mathrm{Concept}_i) = \frac{\alpha}{\sqrt{\mathrm{size}(\mathrm{Concept}_i)}}$$

- The sample is density preserving and biased by group size.
  - For a given sample size M, the value of α is given by:

$$\alpha = \frac{\mathrm{M}}{\sum_{i=1}^{649} \sqrt{\mathrm{size}(\mathrm{Concept}_i)}}$$

5. NATURAL LANGUAGE PROCESSING

The PhraseX program developed at the National Library of Medicine is used to extract Noun Phrases from the documents. PhraseX extracts noun phrases from text by referring to the syntactic structure provided by the SPECIALIST minimal commitment parser. The SPECIALIST minimal commitment parser relies on the SPECIALIST Lexicon as well as the Xerox stochastic tagger [41]. The output contains simple noun phrases. The authors in [42] refer to these phrases as "core noun phrase," that is, a noun phrase with no modification to the right of the head.

The SPECIALIST parser is based on the notion of barrier words [43] which indicate boundaries between phrases. After lexical look-up and resolution of category label ambiguity by the tagger, complementizers, conjunctions, modals, prepositions, and verbs are marked as boundaries. Subsequently, boundaries are considered to open a new phrase (and close the preceding phrase). Any phrase containing a noun is considered to be a (simple) noun phrase, and in such a phrase, the right-most noun is labeled as the head; all other items (other than determiners) are labeled as modifiers. An example of the output from the SPECIALIST parser is given in (2) for the input in (1).

```
(1) Kupffer cells from halothane-exposed guinea
    pigs carry trifluoroacetylated protein
    adducts.

(2)[[mod([lexmatch(['Kupffer']),
          inputmatch(['Kupffer']),tag(noun)]),
    head([lexmatch([cells]),
    inputmatch([cells]),tag(noun)])],
   [prep([lexmatch([from]),
          inputmatch([from]),tag(prep)]),
    mod([lexmatch([halothane]),
         inputmatch([halothane]),tag(noun)],
         punc([inputmatch([-])]),
         mod([lexmatch([exposed]),
              inputmatch([exposed]),tag(adj)]),
         head([lexmatch(['guinea pigs']),
               inputmatch([guinea,pigs]),
                          tag(noun)])],

   [verb([lexmatch([carry]),inputmatch([carry]),
                                  tag(verb)])],
   [mod([lexmatch([trifluoroacetylated]),
         inputmatch([trifluoroacetylated]),
                             tag(adj)]),
    mod([lexmatch([protein]),
```

```
      inputmatch([protein]),tag(noun)]),
  head([lexmatch([adducts]),
        inputmatch([adducts]),tag(noun)]),
  punc([inputmatch(['.'])])]]
```

The underspecified structure produced by the SPECIALIST parser serves as the basis for the extraction of noun phrase strings by PhraseX. In addition to the simple noun phrase (labeled as "simp" in output), PhraseX identifies two additional structures. One of these is the complex noun phrase in which a head is followed by contiguous prepositional phrases to its right ("macro"). The first preposition in this structure can be anything, but all the rest must be "of". The second structure is not a canonical syntactic phenomenon, but may be important for information processing. Such a phrase includes all the content words that occur in a sentence either to the left or the right of a finite verb ("mega"). Examples of these strings as extracted from the syntactic structure in (2) are given in (3).

```
(3) 00000000|simp|kupffer cells
    00000000|simp|halothane exposed guinea pigs
    00000000|simp|trifluoroacetylated protein
                                          adducts
    00000000|macro|kupffer cells from halothane
                              exposed guinea pigs
    00000000|mega|kupffer cells from halothane
                              exposed guinea pigs
    00000000|mega|trifluoroacetylated protein
                                          adducts
```

6. DOCUMENT INDEXING

There are two possibilities related to indexing the documents:

- Terms are used as dimensions of the underlying vector space as in the SMART Indexing and Retrieval Engine [44].
- The Latent Semantic Indexing approach [47][52], where a *Singular Value Decomposition (SVD)* analysis identifies the underlying eigenvectors. These are used as dimensions of a common "latent" space in which both term and document vectors can be represented.

Either technology can be used with either words or phrases as features. The documents can be pre-processed to extract noun phrases, which can then be indexed by using either of the above approaches. Alternatively, the raw text bag of words, after removal of stop words, can be indexed.

Singular Value Decomposition 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 [52]. 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 [52] provide a formal justification for using the matrix of left singular vectors $U_k(E)$ as a vector lexicon.

7. CLUSTERING THE DATA SET

The document vectors generated by the document indexing engine undergo a clustering process, using a bisecting k-means algorithm. A hierarchical cluster tree is generated. Consider a set of document vectors $\boldsymbol{D} = \{d_1, \ldots, d_M\}$ in the Euclidean space $\mathbf{R}^N$. Let the centroid of the set be denoted by:

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

The cohesiveness of the set (also known as intra-cluster cohesiveness) is defined as:

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

Let $\{\pi_i\}_{i=1}^{k}$ be a partition of $\boldsymbol{D}$ with the corresponding centroids $m_1 = m(\pi_1), \ldots, m_k = m(\pi_k)$. The quality of the partition increases if the intra-cluster cohesiveness increases. Thus the quality $\boldsymbol{Q}$ of the partition $\{\pi_i\}_{i=1}^{k}$ is given by:

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

We start with the set of all the documents as the initial cluster. Let $C_1, \ldots, C_i$ be the set of clusters at i^{th} iteration. We choose a cluster S using a selection rule and apply k-means clustering with $k=2$ to give $(i+1)$ clusters. Typically a cluster with the lowest intra-cluster cohesiveness or the one with maximum intra-cluster variance is chosen. We check to determine if there is significant improvement in the partition quality. In case there is, we run k-means on all the $(i+1)$ clusters to stabilize the clusters at this level. Changes in the clusters are noted and the above process is repeated until a significant increase in the quality measure is not seen. The algorithm pseudo-code is presented below.

```
1.  Start with a single cluster D at level = 1.
2.  At tree level = L,
    a.  Select a cluster π_{j,L} from the partition
        {π_{i,L}}^k_{i=1} which has the lowest value for
        c(π_{j,L})
    b.  Run k-means clustering on {π_{j,L}} with k = 2
        to obtain a new partition with k+1
        clusters {π_{i,L+1}}^{k+1}_{i=1}. This includes the
        clusters {π_{j,L+1}, π_{k+1,L+1}} generated from
        cluster π_{j,L}.
3.  Check if Q({π_{i,L+1}}^{k+1}_{i=1}) is significantly
    greater than Q({π_{i,L}}^k_{i=1})
4.  If there are significant gains,
    a.  Copy the centroids to initialize a new
        partition at level L+1, i.e., m_i = m(π_{i,L+1})
    b.  Establish the following relationships:
        i.   child(π_{j,L}) = π_{j,L+1}
        ii.  child(π_{j,L}) = π_{k+1,L+1}
        iii. child(π_{i,L}) = π_{i,L+1} for other clusters.
    c.  Run k+1 means clustering on {π_{i,L+1}}^{k+1}_{i=1} to
        stabilize the clusters at level L+1
    d.  Goto step 2.
5.  Stop.
```

It should be noted that the hierarchical cluster tree is an artifact of the clustering algorithm and is **not** the taxonomy that will be generated. As a part of the clustering process, we compute certain parameters that will be useful in extracting the final taxonomy. The parameters are:

- The intra-cluster cohesiveness $c(\pi_i)$. This determines the differentiation in meaning between successive levels of the extracted taxonomy.
- The centroid vector $m(\pi_i)$. This is used to generate potential labels corresponding to a cluster.
- The parent child relationships between the clusters generated at the various levels.

The clustering strategy involves design choices discussed below:

- **Document *vs* term clustering:** Document clustering is preferred over term clustering, as in most real data sets there are more terms than documents, giving the clustering algorithm a greater discerning power to differentiate clusters.
- **Avoiding Local Extrema:** We adopt two strategies to avoid the clustering process from getting caught in a local extrema:
  - The clustering process is initiated by generating random seed centroid and then performing K-means iterations until convergence is reached or a maximum number of iterations have been performed (Step 2b of the algorithm). A seed is generated from a different part of the vector space and the best partitioning (based on the quality measure discussed) is chosen across multiple K-means runs.
  - The clustering by initializing the centroids from the 2-means step and performing a K-means run at each stage (Step 4c of the algorithm).

8. TAXONOMY EXTRACTION

According to our taxonomy extraction hypothesis, *nodes at lower levels in the taxonomy should capture subject categories that correspond to a narrower information space as compared to nodes at higher levels, and successive levels in the taxonomy should be sufficiently differentiated to be of interest to the user*. The notion of differentiation is captured by the difference in the ***cluster cohesiveness*** between successive layers of the hierarchical cluster tree. 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. In the course of our experimentation, it was observed that the successive values of cohesiveness down a cluster hierarchy are ***monotonically increasing*** in value. In general, this will be an iterative process involving display of the raw clustering and labeling results to the user. This will give him/her a better idea of how to set up the cohesiveness levels to produce the desired taxonomy. The levels of cohesiveness are thus parameters which can be varied to better "tune" a taxonomy that corresponds to the creator's perspective of the information domain. The process of interaction between the taxonomy creator and the TaxaMiner system and "tuning" of the parameters are beyond the scope of this paper and will be addressed in our future work.

Given a set of cohesiveness parameters, the taxonomy extraction algorithm extracts a subset of nodes from the clustering hierarchy and identifies the taxonomic structure (**Figure 2**). The input to this algorithm is a cluster hierarchy (H) with the computed cohesiveness measure $c(\pi_i)$ and a set of thresholds: $\mu_1 \geq \ldots \geq \mu_N$ and the output is an extracted taxonomy (T).

A set of paths belonging to a tree T is denoted by $paths(\boldsymbol{T}) = \{p_1, \ldots, p_M\}$ and contains the paths originating from the root of the tree and ending at the leaf nodes of the tree. The paths corresponding to the hierarchical cluster ***H*** in **Figure 2** are:
$paths(\boldsymbol{H}) = \{$"DSSS…", "$DHH_4\ H_4\ H_4$...", "$DHKLH_3$…", …$\}$.

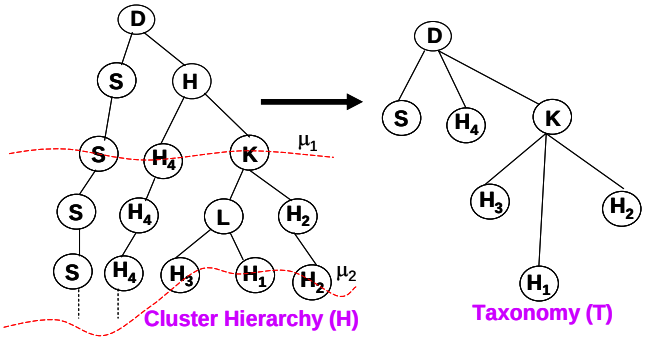

Figure 2: Taxonomy Extraction from Hierarchical Cluster Tree

Each node in ***H*** corresponds to a cluster of documents. A set of selected nodes corresponding to a cohesiveness threshold μ_j is denoted by $selectedNodes(\mu_j)$ and identifies clusters π_j s.t. $c(\pi_j)$ is closest to μ_j. The selected nodes as illustrated in **Figure 2** are:

$selectedNodes(\mu_1) = \{S, H_4, K\}$
$selectedNodes(\mu_2) = \{ H_3, H_1, H_2\}$

We now present an algorithm for taxonomy extraction.

```
1.  For each path pi in paths(H) do
    a.  For j = 1 to N do
        i.  Find nodes A and B in pi s.t. c(A) ≤ μj
            ≤ c(B)
       ii.  If (μj - c(A)) ≤ (c(B)-μj)
            Insert A in selectedNodes(μj)
            Else, Insert B in selectedNodes(μj)
2.  Collapse H: For i = 1 to N do
    a.  For each Node A in selectedNodes(μi) do
        i.  If i>1,
            Find ancestor(A) in selectedNodes(μi-1)
       ii.  If i=1, ancestor(A) = root(H)
      iii.  Delete all nodes from on the path from
            A to ancestor(A)
       iv.  Establish ancestor(A) as the parent of
            A in the extracted taxonomy T
3.  End Extract Taxonomy
```

9. TAXONOMY LABELING

Once the relevant taxonomy nodes have been extracted from the cluster hierarchy tree, the following steps are performed:

- For each node in the extracted taxonomy, a set of potential labels that are extracted.
- These sets of labels are then refined using two main techniques: taxonomic propagation and term neighborhood expansion.

The extraction of the top K terms that contribute most to the centroid vector in a given node can be implemented in the following two ways:

- In the case of SMART [44], terms and documents have their own underlying vector spaces. Hence, we simply choose the top K values of the centroid vector and determine the terms which contribute to the top K terms.
- In the case of the LSI [47], terms and documents are represented in the same "latent" space. This enables us to compute the (Euclidean or cosine) distance between the centroid vector and the term vectors.

Given a cluster node π_i, we define the *labels(π_i)* to contain the labels assigned to the cluster in the taxonomy tree.

$$\text{childLabels}(\pi_i) = \bigcup_{A \in \text{children}(\pi_i)} \text{labels}(A)$$

$$\text{parentLabels}(\pi_i) = \text{labels}(\text{parent}(\pi_i))$$

$$\text{taxonomyLabels}(\boldsymbol{T}) = \bigcup_{A \in \boldsymbol{T}} \text{labels}(A)$$

9.1 Taxonomic Propagation

Having assigned labels to each of the nodes in the extracted taxonomy, the first challenge is to determine which of the K labels are relevant to the node and which are spurious. Some heuristics for taxonomic label propagation 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 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 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 taxonomic label propagation is as follows.

```
1.  Start with the Root(T)
2.  For each cluster node πi at level L do
    a.  For cluster node πj ∈ children(πi) do
        i.  If Δ = labels(πi) ∩ labels(πj) ≠ ϕ
       ii.  labels(πi) = labels(πi) - Δ
3.  End Propagate to Children
4.  Start with cluster nodes in leaves(T)
5.  For each cluster node πi at level L do
    a.  If Δ = labels(πi) ∩ childLabels(πi) ≠ ϕ
    b.  labels(πi) = labels(πi) + Δ
    c.  For πj ∈ children(πi) do
        i.  labels(πj) = labels(πj) - Δ
6.  End Propagate to Parent
7.  End Label Propagation
```

9.2 Term Neighborhood Expansion

The use of LSI enables application of a technique, referred to as *Term Neighborhood Expansion (TNE)*, which attempts to further reduce the number of potential labels for each node in the final hierarchy. Let *labels(π_i)* represent the labels in a node. Let $l_j \in$ *labels(π_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| t ∈ Max$_k$({$\vec{t}_i$. $\vec{l}_j$ }), t_i ∈ term vector lexicon}*

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,

$$N(\pi_i) = \bigcup_{1 \le j \le n} \text{neighborhood}(l_j)$$

where, $l_j \in \textit{labels}(\pi_i)$, and *n* is the number of labels in node π_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:

$$W(l_m) = \sum_{1 \le j \le n} w(l_m) \text{ where } l_m \in \textit{neighborhood}(l_j)$$

and $w(l_m) = |\vec{l}_m \bullet \vec{l}_j|$

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

10. EXPERIMENTAL EVALUATION

We now discuss metrics used to evaluate the quality of the taxonomy generated by our algorithms. Experiments that investigate the impact of the various components of the framework discussed in Section 3 on the quality of the taxonomy generated are discussed.

10.1 Taxonomy Quality Metrics

We propose to separate the content and structural aspects of a taxonomy, in an attempt to discover trade-offs and dependencies that might exist between the two. This will enable us to determine which of the steps in our process contributes to an increase in the quality of the taxonomy generated. Towards this end, we propose simple and pragmatic metrics to evaluate the generated taxonomy *wrt* a gold standard taxonomy. There are two classes of metrics: those that measure the quality of the content (labels); and those that measure the structure of the generated taxonomy.

Content Quality Metric (CQM): This measures the overlap in the labels present in the generated Taxonomy, $\boldsymbol{T_{gen}}$ and the gold standard taxonomy $\boldsymbol{T_{gold}}$. There are two variants of this metric:

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

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

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

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

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

$$pcLinks(\boldsymbol{T}) = \{<a,b> \mid a \text{ is parent of } b \text{ in } \boldsymbol{T}\}$$

$$adLinks(\boldsymbol{T}) = \{<a,b> \mid a \text{ is ancestor of } b \text{ in } \boldsymbol{T}\}$$

$$adLinks(\boldsymbol{T}) \supseteq pcLinks(\boldsymbol{T})$$

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

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

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

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

10.2 Experimental Results

We present an initial set of experiments evaluating the impact of the following on the quality of the taxonomies generated.

- The effect of varying the size of the data sets.
- The effect of varying the number of labels extracted.
- The effect of pre-processing the document set using limited NLP techniques (Noun Phrase Extraction)
- The effect of using Latent Semantic Indexing (Section 6) and Term Neighborhood Expansion (Section 9.2)

In our approach, a subject matter expert is required to set the threshold levels for taxonomy extraction, i.e., the μ values discussed in Section 8. However, current experiments reflect μ values assigned automatically based on the minimum and maximum values of cohesiveness, and the involvement of an expert would significantly improve the quality measures. Also, it may be noted that our techniques will not be able to generate labels in the taxonomy that do not appear in the text of the MEDLINE® abstracts. The gold standard taxonomy and examples of generated taxonomies are illustrated in the appendix at the end of this paper.

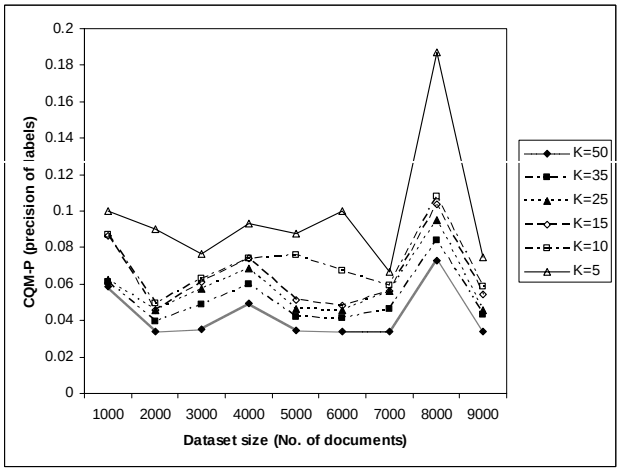

Figure 3: Content Quality Metric (Precision)

Figure 3 above, illustrates the impact of using datasets of different sizes on CQM-P. It may be noted that ncreasing the data set size does not necessarily increase the values of CQM-P. In fact, we notice a trend that suggests that CQM-P peaks for a certain value of the data set size and then deteriorates for larger data sets. We observe this behaviour across all values of *K* (the number of labels extracted). Also, extracting a lesser number of labels for each cluster node (the value of *K*) gives better results for CQM-P

Figure 4 below, illustrates the impact of using datasets of different sizes on CQM-R. It may be noted that the value of CQM-R stays in a narrow band (0.5 – 0.6) and appears to be relatively independent of the size of the dataset and the number of labels extracted.

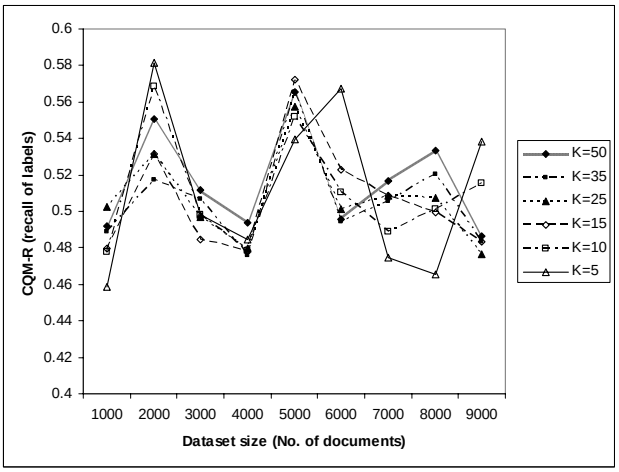

Figure 4: Content Quality Metric (Recall)

Figure 5 below illustrates the impact of using datasets of different sizes on SQM-P. In contrast to the content quality metric CQM-P, increasing the value of *K* (the number of extracted labels), gives better values of SQM-P. More interestingly, the values of SQM-P show a downward trend *wrt* dataset size, i.e., increasing the size of the dataset, results in a decrease in SQM-P.

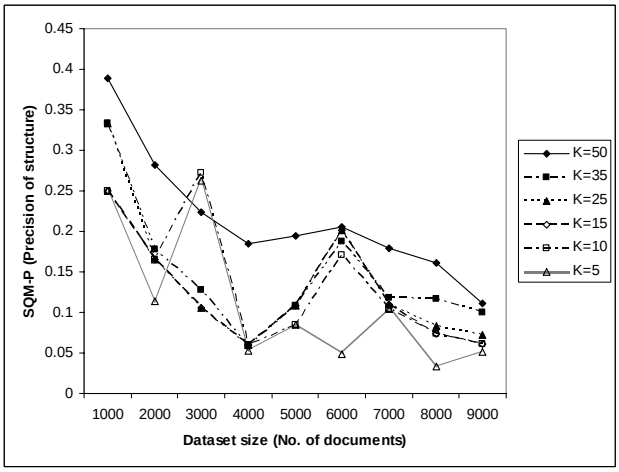

Figure 5: Structural Quality Metric (Precision)

Figure 6 below, illustrates the impact of using datasets of different sizes on SQM-R. In a manner similar to SQM-P, SQM-R shows a downward trend on increasing the dataset size, i.e., as the data set size increases, SQM-R tends to decrease. Also, similar to SQM-P again, extracting a larger number of labels (value of *K*), tends to increase SQM-R, notwithstanding a few "crossings".

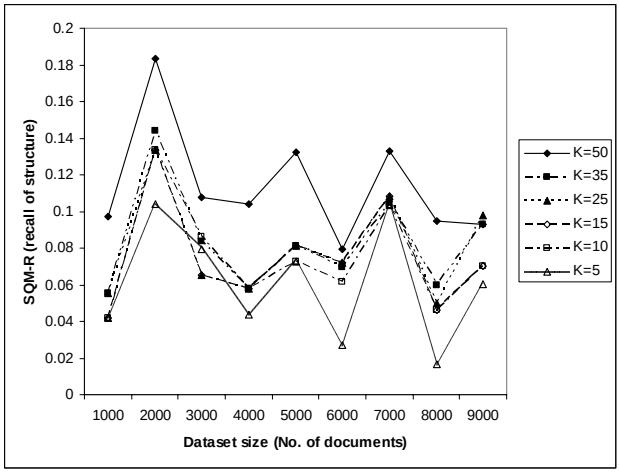

Figure 6: Structural Quality Metric (Recall)

In the next set of experiments, we investigate the impact of pre-processing the document set using limited NLP techniques, such as noun phrase extraction.

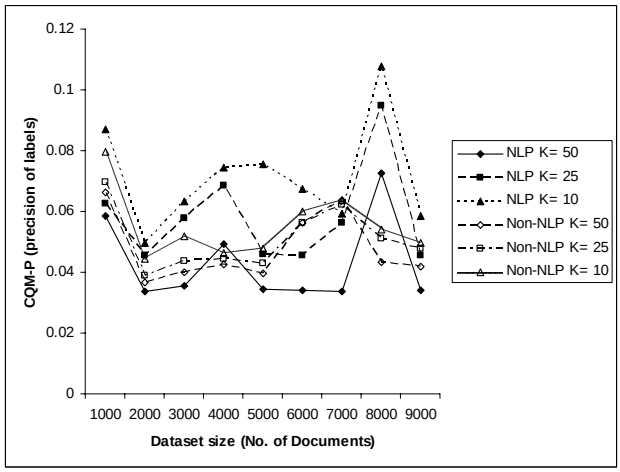

Figure 7: NLP vs. Non-NLP for CQM-P

Figure 7 above compares the impact of using NLP techniques vis-à-vis not using them on the precision-based content quality measure. We observe that for each value of *K* (the number of labels extracted), the values for CQM-P are consistently better for the NLP case in comparison to the non-NLP case.

Figure 8 below compares the impact of using NLP techniques vis-à-vis not using them on the recall-based content quality measure. We observe that for each value of *K* (the number of labels extracted), the values for CQM-R are consistently better for the non-NLP case in comparison to the NLP case. This is an interesting trend in complete contrast to the trend observed in the case of CQM-P.

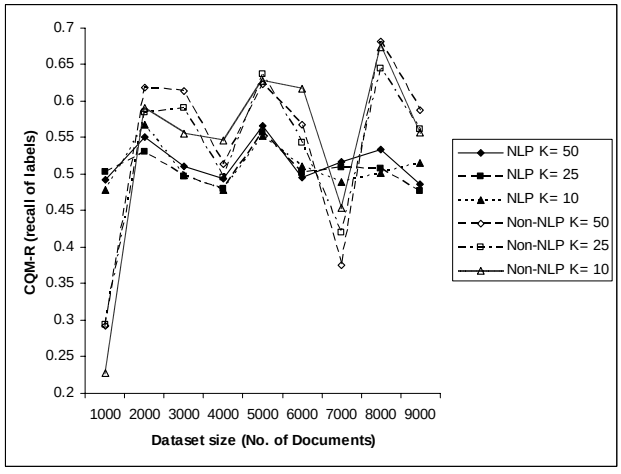

Figure 8: NLP vs. non-NLP for CQM-R

Figure 9 shows the values of 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 10** 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 11** 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 topic hierarchy independent of the number of labels extracted.

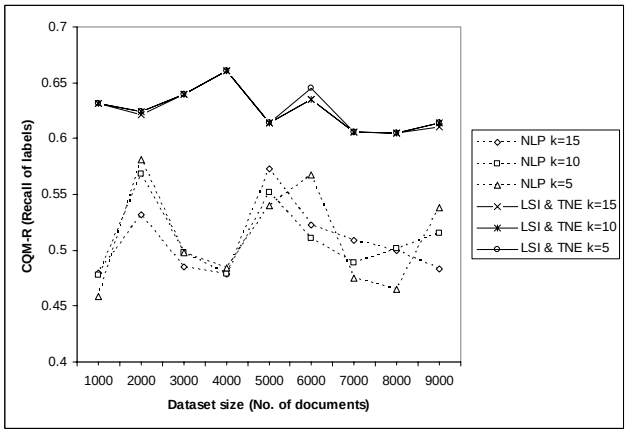

Figure 9: Comparison of Content Quality Recall (LSI+TNE *vs.* SMART+NLP)

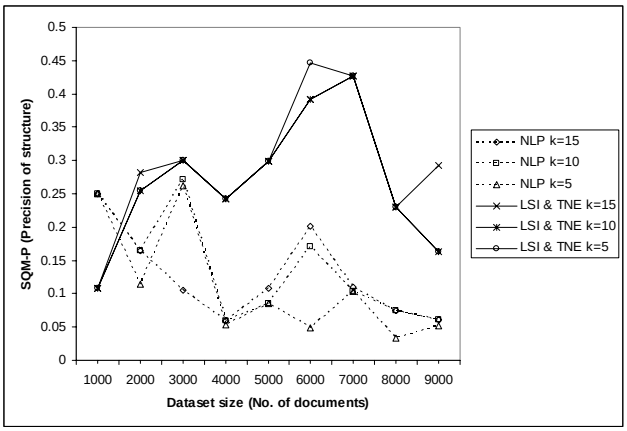

Figure 10: Comparison of Structure Quality Precision (LSI+TNE vs. SMART+NLP)

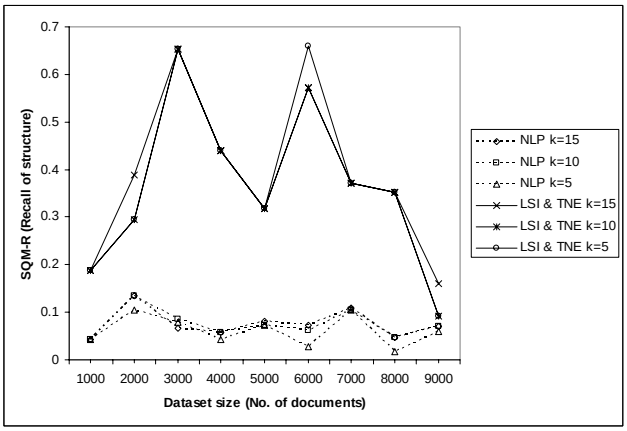

Figure 11: Comparison of Structure Quality Recall (LSI+TNE vs. SMART+NLP)

10.3 Discussions and Insights

The experiments discussed above are a component of extensive ongoing work in evaluating a suite of taxonomy generation techniques. They have provided us with some interesting insights, which indicate further areas of research and investigation. Assuming that in general, the goal of a taxonomy designer will be to optimize both the content and structural quality of a taxonomy, the trends observed in the previous section point to an optimal size of the data set and an optimal number of labels to be extracted at each node. Some relevant observations in the previous section are:

- There is a trend for CQM-P to peak for a particular size of the data set and this trend is visible across all values of *K* (number of labels extracted).
- Extracting a lesser number of labels (value of *K*) tends to increase the values of CQM-P.
- The values of SQM-P on the other hand (in contrast to CQM-P) tend to improve for a higher number of labels extracted (value of *K*).
- Also, in contrast to CQM-P, the values of SQM-P decrease with an increase in the size of the dataset.
- The values of SQM-R, in a manner similar to SQM-P also tend to increase with an increase when the dataset size decreases.

The low values of the various quality measures (except CQM-R) suggest that a deeper investigation is needed to obtain better results. We expect meaningful user input in the form of judiciously chosen cohesiveness thresholds (μ values) to alleviate the problem by identifying the correct level of differentiation and alignment.

The use of LSI in conjunction with TNE has increased the quality of the taxonomies being generated (**Figure 9-11**). LSI helps identify latent salient "concepts" in the corpus based on which term and document vectors are constructed. 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. This helps in restricting the labels of node to the salient domain terms, resulting in the increase in the quality of taxonomies generated.

11. CONCLUSIONS AND FUTURE WORK

The main contribution of this paper is a comprehensive approach and framework for the difficult, and yet important problem for bootstrapping taxonomies from textual data. In contrast to other approaches, that address components of the problem, we present a comprehensive process and strategy that minimizes the involvement of a domain expert in creating a taxonomy. Some of the novel features of our work are:

- A systematic experimental framework that combines and evaluates statistical clustering, NLP, LSI and other techniques for taxonomy generation. Design of taxonomy quality metrics and their use to evaluate the impact of the above techniques on the quality of the results generated.
- Exploitation of the statistics generated during the clustering process to extract a more meaningful taxonomy. Identification of statistical parameters that characterize the notion of "differentiation" in the taxonomic structure.
- Techniques for automatic generation and refinement of labels for creating the final taxonomy. Use of LSI and TNE in this context that results in a dramatic improvement in the quality of the taxonomy generated.
- Initial validation of our approach using a real world data set, the MEDLINE® database and real world taxonomy, the MeSH thesaurus.

Human involvement, though minimized is crucial to the process of creating good quality taxonomies. Also, taxonomy quality is a combination of content-based and structure-based components which can be weighted differently to reflect different application characteristics. Finally, an optimal strategy for taxonomy generation based on a user configured quality metric involves a joint optimization of various parameters. Some issues that we are investigating in the context of ongoing work are:

- Algorithmic techniques for improving the structural quality of the generated taxonomies.
- Understand and leverage the human expert, especially in the context of identifying the levels of differentiation in the taxonomy that corresponds to his/her perspective of the application or domain. Combined quality metrics that better reflect the needs of the user.
- Investigation of the notion of an optimal set of parameters for generating a taxonomy. For example, processing a bigger data set can be avoided if we know that the resulting improvement in the taxonomy quality will be negligible.

- Investigation of NLP and other techniques [7] to further refine the taxonomies generated into richer ontologies.

We believe that pragmatic issues as enumerated above are crucial for generating ontologies/taxonomies in a scalable and feasible manner and that we have taken a very important first step in this direction.

12. REFERENCES

[1] T. Berners Lee, J. Hendler and O. Lassila. The Semantic Web. *Scientific American*, May 2001.

[2] J. Kahan, M-R. Koivunen, E. Prud'Hommeaux and R. Swick. Annotea: An open RDF Infrastructure for shared annotations. *Proceedings of the 10th International WWW Conference (WWW 2002)*, Hong Kong, May 2001

[3] S. Handschuh, S. Staab and R. Volz. On Deep Annotation. *Proceedings of the 12th International WWW Conference (WWW 2003)*, Budapest, Hungary. May 2003.

[4] S. Dill et. al. SemTag and SemSeeker: Bootstrapping the Semantic Web via automated semantic annotation. *Proceedings of the 12th International WWW Conference (WWW 2003)*, Budapest, Hungary, May 2003.

[5] MeSH. Medical Subject Headings. Bethesda (MD): National Library of Medicine, 2003. http://www.nlm.nih.gov/mesh/meshhome.html

[6] G Salton, Editor. The SMART Retrieval System – Experiments in Automatic Document Retrieval. Prentice Hall Inc., Englewood Cliffs, NJ 1971.

[7] Hearst, M.: *Automatic acquisition of hyponyms from large text corpora*. In Proceedings of the 14th International Conference on Computational Linguistics. Nantes, France, 1992.

[8] C Y Chung, R. Lieu, J. Liu, A. Luk, J. Mao and P. Raghavan. Thematic Mapping – From Unstructured Documents to Taxonomies. *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM 2002)*, McLean, VA, November 2002.

[9] 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.

[10] 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

[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] P. Jacobs and U. Zernik. Acquiring Lexical Knowledge from Text: A Case Study. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, 1988.

[13] P. Hastings and S. Lytinen. The Ups and Downs of Lexical Acquisition. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 1994.

[14] R. C. Berwick. Learning Word Meanings from Examples. In *Semantic Structures: Advances in Natural Language Processing*. Lawrence Erlbaum Associates, 1989.

[15] C. Cardie. A Case-based Approach to Knowledge Acquisition for Domain Specific Sentence Analysis. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 1993.

[16] D. H. Fisher. Knowledge Acquisition via incremental conceptual clustering. *Machine Learning 2:139-172*, 1987

[17] 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.

[18] B. S. Everitt, S. Landau and M. Leese. Cluster Analysis. Edward Arnold. 4th Edition, May 2001.

[19] Y. Zhang and G. Karypis. Criterion functions for Document Clustering. *Technical Report, U. Minnesota, Dept. of Computer Science, #TR-01-40*.

[20] S. Chakrabarti. Data Mining for Hypertext: A Tutorial Survey. *ACM SIGKDD Explorations*, 1(2):1-11, 2000.

[21] 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.

[22] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. In *Proceedings of ACM SIGIR Conference*, 1998.

[23] 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.

[24] M. Finkelstein-Landau and E. Morin. Extracting Semantic Relationships between Terms: Supervised vs Unsupervised Methods. In *Proceedings of International Workshop on Ontological Engineering on the Global Information Infrastructure*, Dagstuhl Castle, Germany, May 1999.

[25] 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

[26] M. Missikoff, P. Velardi and P. Fabriani. Text Mining Techniques to automatically enrich a Domain Ontology. *Applied Intelligence* 18, 323-340, 2003.

[27] M. Sanderson and B. Croft. Deriving Concept Hierarchies from Text. *International Conference on Research and Development in Information Retrieval (SIGIR 1999)*, 1999.

[28] M. Reinberger, P. Spyns, W. Daelemans and R. Meersman. Mining for Lexons: Applying unsupervised learning methods to create ontology bases.

[29] A. Nazarenko, P. Zweigenbaum, J. Bouaud and B. Habert. Corpus-based identification and refinement of semantic classes. In *Proceedings of the AMIA Annual Symposium*, 1997.

[30] D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL-98*, 1998.

[31] A. Maedche and S. Staab. Discovering conceptual relations from text. Technical *Report 399, Institute AIFB, Karlsruhe University*, 2000.

[32] 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.

[33] A. Borgida and P. F. Patel-Schneider. A semantics and complete algorithm for subsumption in the CLASSIC description logic. *AT&T Technical Memorandum*, 1992.

[34] A. Maedche, G. Neumann and S. Staab, Bootstrapping an Ontology Based Information Extraction System. *Studies in Fuzziness and Soft*

Computing, INTELLIGENT EXPLORATION OF THE WEB, P.S. Szczepaniak, J. Segovia, J. Kacprzyk, L.A. Zadeh, Springer, 2003.

[35] 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.

[36] A. Maedche and S. Staab. Ontology learning for the Semantic Web. *IEEE Intelligent Systems*, 16, 2001.

[37] 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

[38] C. Buckley, M. Mitra, J. Walz and C. Cardie. Using clustering and superconcepts within SMART: TREC 6. In *Sixth Test Retrieval Conference (TREC-6)*, Gaithersburg, MD, November 1997.

[39] 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.

[40] E. Rasmussen. Clustering Algorithms. In W. B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, 1992

[41] D. R. Cutting, J. Kupiec, J. O. Pedersen and P. Sibun. A practical part-of-speect tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, 1992.

[42] R. Weischedel, M. Meteer, R. Schwartz, L. Ramshaw and J. Palmucci. Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, 19(2), 1993.

[43] K. W. F. Tersmette, A. F. Scott, G. W. Moore and R. E. Miller. Barrier word method for detecting molecular biology multiple word terms. *Proceedings of the 12th Annual Symposium on Computer Applications in Medical Care*, 1988

[44] G. Salton. The SMART Retrieval System – Experiments in Automatic Document Retrieval, Prentice Hall, 1971.

[45] J. T. Wang, K. Zhang, K. Jeong and D. Shasha. A System for Approximate Tree Matching. *IEEE Transactions on Knowledge and Data Engineering*, 6(4), August 1994.

[46] Y. Park, R. Byrd and B. Boguraev. Towards Ontology on Demand. *Proceedings of the ISWC Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*, October 2003.

[47] Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A., "ndexing by latent semantic analysis. *Journal of the Society for Information Science, 41(6)*, 391-407, 1990.

[48] 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, December 2002, pp. 29-49.

[49] 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.

[50] D. J. Lawrie and W. B. Croft. Generating hierarchical summaries for web searches. In Proceedings of SIGIR, pages 457 -- 458, 2003.

[51] D L Boley. Principal Direction Divisive Partitioning, *Data Mining and Knowledge Discovery, Volume 2(4)*, 1998.

[52] Berry, M., Dumais, S., and O'Brien, G. 1995. Using Linear Algebra for Intelligent Information Retrieval. SIAM Review, 35(4).

Appendix

Figure 2 Example of Learnt Hierarchy (note that the darker shaded nodes and capitalized labels indicate a match with a gold taxonomy node)

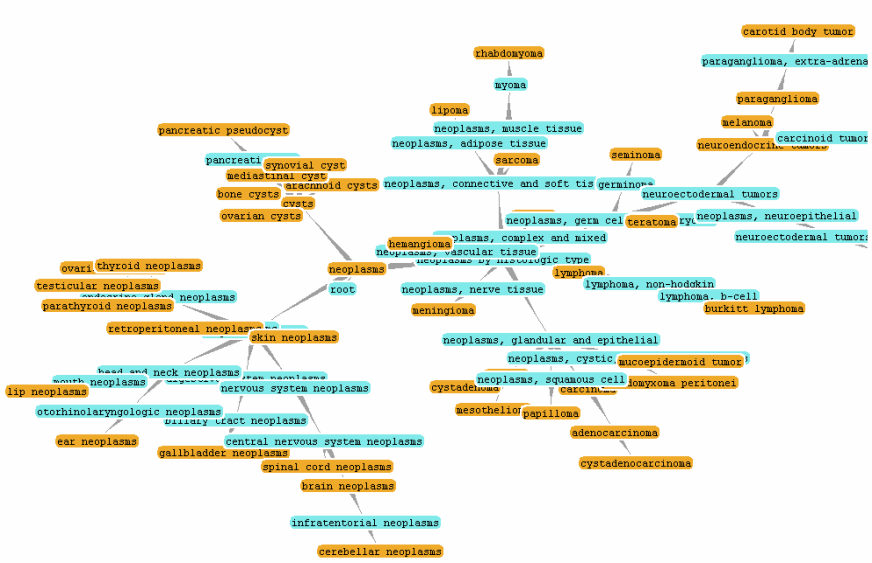

Figure 3 Corresponding portion of MeSH (gold standard hierarchy)