Multiple Hierarchical Classification of Free-Text Clinical Guidelines

Robert Moskovitch, Shiva Cohen-Kashi, Uzi Dror, Iftah Levy, Amit Maimon and Yuval Shahar

Medical Informatics Research Center

Department of Information Systems Engineering, Ben Gurion University

P.O.B. 653, Beer Sheva 84105, Israel

Corresponding Author:

Robert Moskovitch

Medical Informatics Research Center

Department of Information Systems Engineering

Ben Gurion University

P.O.B. 653, Beer Sheva 84105, Israel

Tel: +972-52-2668071, Fax: +972-8-6477160

email: robertmo@bgu.ac.il

Abstract

Objective: *Manual classification of free-text documents within a predefined hierarchy, commonly required in the medical domain, is highly time consuming task. We present an approach based on supervised learning to automate the classification of clinical guidelines into predefined hierarchical conceptual categories.*

Methods and Material: *Given a set of hierarchically categorized documents in the training stage the learning algorithm exploits the hierarchical structure of the concepts in order to overcome the low number of training examples. The classification task is thus decomposed into a continuous decision process, not unlike searching within a decision tree, which follows the concept hierarchy and which needs to make a decision at each node on the path. Classification is based on applying a similarity function at each concept. Several evaluation measures were used, based on the intended use of the hierarchy. In addition, conservative and aggressive stop-criterion strategies for stopping the search through the concept hierarchy were formulated. An evaluation of the approach, including several training methods and multiple evaluation measures, has been performed using a training set of 1136 guidelines from the National Guideline Clearing House set.*

Results: *Based on a test collection consisting of 1038 CPGs classified along two hierarchies, of roughly 5,000 concepts, in which each CPG was classified* *by a mean of 10 concepts, a variable precision was observed from 44% to 60% depending on the settings of the training methods.*

Conclusion: *These results demonstrate the feasibility of the approach, especially when considering the low ratio of guidelines to classification indices (concepts) in the evaluation data set used here.*

Keywords:

Hierarchical Text Categorization, information retrieval, concept based search, clinical guidelines.

1. Introduction

Recent trends in knowledge management highlight the interest in organizing sources of knowledge or documents within hierarchies of concepts or categories. Web directories are a well known example. Web pages are classified with respect to a predefined set of categories organized into concepts hierarchy, also called a *taxonomy*. Well known examples of such knowledge management include Google (http://directory.google.com, Accessed: 20 March 2006), Yahoo! (http://www.yahoo.com, Accessed: 20 March 2006) and the Open Directory Project (http://www.dmoz.org, Accessed: 20 March 2006). A hierarchical knowledge organization is defined by two components: a hierarchy of categories, implemented in a tree-like structure, and a collection of information items; we will focus on free-text documents. Documents can be classified by one or more concepts at different levels of the hierarchical conceptual structure. Thus, a *concept* is, in reality, a *path* in the conceptual hierarchy, leading to either an internal node or a leaf node in the concepts tree. Documents are classified mainly manually by experts or by using scripts that represent classification heuristics; these are prepared manually as well, and are hard to define. Another option is automated text classification using machine learning techniques. We suggest combining this approach with the manual approach, in which the expert manually classifies the documents. Thus, the automated tool recommends a set of concepts and the expert can modify the concepts by removing and adding concepts. This combination is expected to reduce the expert's efforts and overcome his lack of familiarity with the taxonomy concepts (hierarchies may include thousands of concepts) as the expert has only to confirm or modify the recommendations.

Much has been done in the field of flat Text Categorization [1], while more could be accomplished in the field of Hierarchical Text Categorization. We will describe an automated, machine learning approach to document classification, focused on the domain of clinical practice guidelines. First we describe the motivation for this project. We describe the Digital Electronic GuidelinE Library (DeGeL); then we describe Vaidurya, the search engine of DeGeL, implementing the concept-based search, which requires the automated multiple classification of clinical guidelines in a hierarchy of concepts. We describe the classification algorithm and an evaluation we performed. Finally we discuss the results and their conclusions.

1.1. Clinical Practice Guidelines and The DeGeL Architecture

Clinical practice guidelines (**CPGs**) are an increasingly common and important format in clinical medicine for prescribing a set of rules and policies that a physician should follow. According to studies, CPGs improve medical practice. They improve the quality and possibly also the cost-efficiency of care in an increasingly complex health care environment [2].

To support tasks such as the runtime application of a guideline, it is often important to be able to retrieve quickly a set of guidelines most appropriate for a particular patient or task. Correctly classifying the guidelines, along as many semantic categories as relevant (e.g., therapy modes, disorder types, sighs and symptoms), supports easier and more accurate retrieval of the relevant guidelines. This classification, however, is mostly manual. Electronic CPG repositories, such as the **National Guideline Clearinghouse** (**NGC**) (http://www.ngc.org, Accessed: 20 March 2006) and others provide hierarchical access to electronic CPGs in a free-text or semi-structured format.

Ideally, automated support could be offered to guideline-based care at the point of care. There is a global effort to automate the representation and application of CPGs [3]. To convert CPGs gradually into a machine executable representation in a target ontology, we have developed a web-based, distributed architecture, the **Digital Electronic GuidelinE Library**, (**DeGeL**) [4], and a set of software tools for incremental conversion of CPGs through semi-structured and semi-formal representations, into one or more formal guideline-specification representations. The editing tools, such as for semantic classification of the whole guideline, and semantic markup of parts of its text by the knowledge roles of the target ontology, are intended for a collaborative effort among medical domain experts and knowledge engineers.

1.2. Classifying Guidelines in the DeGeL Library

To classify guidelines, mainly for purposes of efficient retrieval, seven semantic axes are implemented in DeGeL. Each Axis represents a major clinical aspect. Axes include: (1) symptoms and signs (e.g., hypertension), (2) diagnostic findings (e.g., blood cells count, electrocardiogram), (3) disorders (e.g., ischemic heart disease, malignant neoplasm), (4) treatments (e.g., antibiotic therapy , abdominal surgery), (5) body systems and regions, or a relevant CPG classification, (6) guideline types (e.g., screening, prevention), and (7) guideline specialties (e.g., radiology, internal medicine). The semantic axes in DeGeL are based on the MeSH and UMLS and include concepts which are mostly relevant to CPGs.

Each semantic axis is implemented as a hierarchical structure of subaxes that represent concepts. Each CPG is indexed along one or more semantic axes, such as Disorders (e.g., malignant skin melanoma), Guideline specialties (e.g., oncology), etc. Figure 1 shows an example of a hierarchical concept structure, where the main concept is Mental Disorders. There are two sub-concepts: Anxiety Disorders and Dissociative Disorders. Each sub-concept may be split into more sub-concepts, which represent a more detailed division of the concept. (A semantic indexing structure exists, albeit in a less developed form, also in the NGC repository, using two main Axes based on the inherently hierarchical MeSH and UMLS classification [5,6]: *Disorders* and *Therapies*.) Currently, CPGs in DeGeL are classified *manually* by medical experts, using the *IndexiGuide* tool [4]. Since manual classification is a very time consuming and expensive task, there is a need for an automated classification tool that will make accurate classification faster and cheaper.

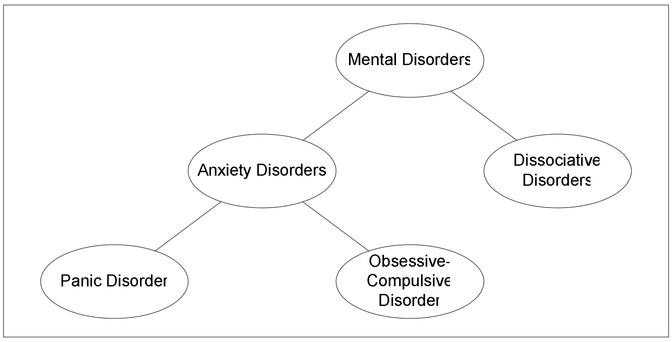

Figure 1 - Part of a hierarchical conceptual structure of the 'Mental Disorders' class

2. The Vaidurya Search Engine

One of the DeGeL architecture's tools is **Vaidurya**, a multiple ontology, concept-based, context-sensitive search and retrieval tool for CPGs [7]. Vaidurya uses three types of search: (1) full-text search, based on key terms; (2) concept-based search, which uses the semantic–axes indexing structure; and (3) context-sensitive search, which exploits the semantic markup performed on guidelines in the DeGeL library. Context-sensitive search focuses on searching for key terms only within the scope of a particular ontology-specific semantic knowledge role, such as "eligibility conditions".

The search option in Vaidurya relevant to this study is the concept based search, which enables specification, within a query of one or more semantic axes or subaxes (concepts), and logical operators defining the relations between the axes. The concept search query is thus a collection of constraints represented by chosen subaxes, and the logical relation between them: conjunction or disjunction. When a query mentioning a logical combination of semantic classification axes is executed, the CPGs classified to each concept and its descendents are retrieved. Eventually, based on the logic relations between the concepts (and additional constraints specified in the query, such as finding certain terms only within a particular context, such as *entry conditions*) a subset of CPGs are retrieved, and a rank is calculated for each CPG.

Recent formal evaluation of Vaidurya has shown significant improvement in precision and recall when concept-based search is used with free-text search when compared to using free-text search alone [8]. Specifically, an average increase of 10% in precision was observed.

Our research has therefore focused on determining the appropriate methods for providing automatically reasonable semantic classifications for CPGs, either for supporting an automated Web crawler that suggests additional candidates to add to repositories such as the DeGeL library, or, in a semi-automated fashion, for presenting a default option to an expert editor who is marking-up a guideline.

3. Hierarchical Text Classification

A wide range of statistical and machine learning techniques have been applied to text categorization, including multivariate regression models [9], nearest neighbor [10], Probabilistic Bayesian models [11], decision trees [12], neural networks [13], and support vector machines

[10]. These techniques are all based on having some initial set of class-labeled documents, which is used to train an automatic categorical-classification model.

Although many real world digital libraries, or directories, have a complex hierarchical structure (Google, Yahoo!, ODP and NGC), few learning methods capitalize on this structure [14,15], and even fewer focus on the problem of concurrently proposing multiple hierarchically organized concepts as indices for the new document. Most of the approaches ignore the hierarchical structure and treat each concept or class separately, thus in effect 'flattening' the hierarchical structure. A separate binary classifier is then learned, to distinguish each class from all other classes. The binary classifiers can be considered independently, so an item is classified into either no, one, or more than one category. Alternatively, a classifier can be designed to select the (single) best matching class. Such approaches work well on small problems, but they are likely to be difficult to train when there is a large number of concepts and a very large number of features, as is common in text representation, and especially when the desired output includes multiple, non mutually exclusive, classes.

Hierarchical categorization can support different tasks. Concept-based search, which is commonly used in the medical domain [16,7], requires a prior classification of documents within a predefined hierarchical structure of concepts. In the case of CPGs, it is often important to be able to classify the guideline using multiple semantic concepts coming from different semantic branches of the concept hierarchy. Thus, a guideline for treatment of bronchial carcinoma using irradiation might be classified pathologically under malignancies, anatomically under pulmonary diseases, and therapeutically under radiotherapy. Another use is in presenting search results within a predefined hierarchical structure. For example, Pratt demonstrated an improvement of the visualization of search results by dynamically categorizing the search results within a hierarchical structure of meaningful concepts [17].

Several hierarchical-classification methods were previously applied to a number of data sets. Many studies used the Reuters test collection, in which articles were categorized into 135 categories with no hierarchical structure. Koller and Sahami [18] generated abstracted categories of subsets of the Reuters collection. They compared naïve Bayes, and two Bayesian networks classifiers on flat and hierarchical models. They found advantages in the hierarchical structure over the flat when only a small amount of features (10) were used. Ruiz and Srivinsan [19] classified abstracts within the MeSH sub-category *heart*. Dumais and Chen [14] applied SVM classifiers to web content aiming to classify search results dynamically.

Unlike the method we propose, these studies typically did not experiment with the different choices of training methods to be discussed (e.g., bottom-up as well as top-down propagation of documents through the concept hierarchy), and in particular did not consider the multiple-classification issue. In the case of a hierarchical conceptual structure, it is common to decompose the problem into a set of smaller classification problems. Thus, one first classifies using the concepts at the top level, which are less specific, and then progressively through lower levels, drilling down through the classification path. In contrast, we suggest an approach in which the decision is not only made separately for each node (concept class), but also takes into account the potential classification paths throughout the whole given concept hierarchy.

4. Materials and Methods

4.1. Text Representation and Preprocessing

Guideline documents were represented using the vector space model [20] inspired by the Rocchio approach from information retrieval [21], in which each document is represented by a bag-of-words. A vector of terms is created, such that each index in the vector represents the term frequency in the document, commonly known as *term-frequency (tf).* Equation 1 shows the definition of *tf.* Another common representation is *tf-inverse-document-frequency (TFIDF)*, which combines the frequency of a term in the document and in the documents collection as well. Equation 2 shows the definition of *tfidf*, given the *tf* value of the term, *N* – the number of documents in the entire collection, and *n* – the number of documents in which the term appears.

$$tf = \frac{term\ frequency}{\max(term\ frequency\ in\ document)} \qquad (1)$$

$$tfidf = tf * \log(N/n) \quad . \qquad (2)$$

Each document in the collection goes through an indexing process. Document terms are extracted. Stop words (e.g., "and", "not", "the", etc) are removed and each term is stemmed to its root, using the Porter Stemming Algorithm [22]. Both removals of stopwords, in which common terms are removed, and stemming, in which terms are stemmed to their root, are processes that assist in decreasing the amount of terms in the collection, an aspect that we will be discussing again later in the feature selection section.

4.2. Feature Selection

The features in our case are the document terms. The *Feature Selection* operation refers to the process where the dimensions (number of terms in the indexed document collection) are reduced and only the contributing terms are included in the classification task. This process is essential, since in the textual domain the number of dimensions is very big (many thousands of terms). The dimensional reduction process starts with the stopping and stemming procedure which was described earlier. In this study we are using two feature selection techniques. One of these was adjusted to capitalize on the hierarchical structure of the classes (concepts).

4.2.1. Using the Term Distribution (WD2IDF)

As a general heuristic, we are assuming that features with very high or very low presence in the document *collection* will contribute less to, or even harm, the classification task; thus, we remove these terms, thereby reducing the dimensions of the collection. Term presence is represented using the *idf* value of each term in the collection. Another option could be averaged (over all documents) *tfidf* value. The terms are sorted according to their frequency value, and then multiplied by their relative rank when ordered by frequency. Terms with a high frequency value will have a low rank. Terms with computed value above and below a given threshold are removed.

Mutual Information (MI) is a common measure, which assesses the mutual information between a feature (term) and a class (of documents indexed by a concept) in the collection. Features with a low *MI* value are removed. Equation 3 shows the *MI* value calculation, where *c* represents a class and *t* represents a term in the collection.

$$MI(t,c) = \log \frac{P(t \& c)}{P(t) \times P(c)} \qquad (3)$$

4.2.2. Hierarchical Mutual Information (HMI)

Since our goal is to deal with a hierarchical conceptual structure, we investigated an approach in which the hierarchical structure was explicitly considered. The *HMI* of each feature *x* is calculated not only for the documents classified directly to the given concept-class *C*, but also to the documents classified to the concept descendents in the hierarchy. We call this measure *HMI*.

4.3. The Training Method

The model we present implements a supervised learning technique. In the training process, each concept's vector in the hierarchical conceptual structure is computed based on the CPGs classified by the concept (that is, by the whole classification path through the concept hierarchy ending at that concept). Each concept has *n* CPGs classified directly by the concept, and *m* sub-concepts. The total CPGs classified by a specific concept in the hierarchical structure consists of the *n* CPGs directly classified and the number of CPGs classified by each one of its *m* sub-concepts and their descendents. Each concept is represented by a Terms Vector $c^{\rightarrow}$, a collection of term frequencies <tf_1…tf_k>, in which each entry represents the frequency of one of the *k* terms in the entire collection vocabulary. A *concept vector* is a linear combination of all the CPGs, which are represented by a vector of the document terms and their frequencies, classified by that concept and its descendents.

Our methodology explicitly exploits the hierarchical-classification structure of an existing corpus of documents, in this case CPGs. In effect, we enlarge the number of training examples of the higher level concepts, based on the "IS-A" relation among concepts (Classes) in the hierarchy, assuming that if a CPG is classified by a concept, it is classified also by its ancestors. However, as we will show in the following analysis and experiments, placement of training documents within the hierarchy can be achieved both in a top-down and a bottom-up direction, and the difference (advantages and disadvantages) need to be carefully considered and evaluated.

4.3.1. Bottom-up Training

We implemented two training approaches: the first is *bottom-up*, which is a recursive process starting from the hierarchical concept leaves. For each leaf concept, in the hierarchical structure, a *centroid* terms vector is built based on the CPGs classified by it; at higher levels the centroid is built based on the classified CPGs and the sub-concepts centroids. Equation 4 shows how the concept vector is calculated in the *bottom-up* technique. The resulting centroid vector $\vec{C}$ is the weighted sum of (1) the vectorial mean of the *n* term vectors of the *n* CPGs that are classified by the concept, which are represented by a vector $\vec{d}$, and (2) its *m* sub-concepts vectors $\vec{gi}$ (multiplied by *yi*, the number of documents classified by the sub-concept *i*, for normalization purposes). The weight coefficient *w* represents the influence of the sub-concept vectors. In all of the described experiments, the value of *w* was kept as 0.5, since initial experiments did not show any improvement when its value was other than 0.5; see also the discussion.

$$\vec{C} = (1 - w)\frac{\sum_{i=1}^{n} \vec{d}_i}{n} + w\frac{\sum_{i=1}^{m} y_i \vec{g}_i}{\sum_{i=1}^{m} y_i} \qquad (4)$$

4.3.2. Top-Down Training

The other training approach is *top-down*. In this approach, starting with the root of the concept hierarchy, each CPG vector updates the concept vectors within all of its given classification paths. Equation 5 shows the calculation of a centroid $\vec{C}_{\text{new}}$, given its current concept centroid and the new training CPG vector. Each concept centroid is a simple linear combination of the CPGs vectors classified along the concept or its sub-concepts descendents. Thus, the value of the concept vector in iteration *n* (i.e., when the *n*-th guideline percolates through the hierarchy in *top-down* fashion) is a weighted sum of the concept vector formed after the (*n*-1)th iterations and of the *n*th CPG vector.

$$\vec{C}_n = \frac{\vec{C}_{n-1} \cdot (n-1) + \vec{d}_{new}}{n} \qquad (5)$$

The difference between these two approaches is that in the *top-down* approach each CPG classified directly or indirectly by a node is represented only once in that node; that is, its CPG vector is added only once to the concept's centroid vector, when passing through the node and continuing along its respective classification paths. In contrast, in the *bottom-up approach*, a CPG is represented in the *centroid* of each node that directly or indirectly indexes it, as many times as there are sub-concept descendents of the node by which the guideline is classified.

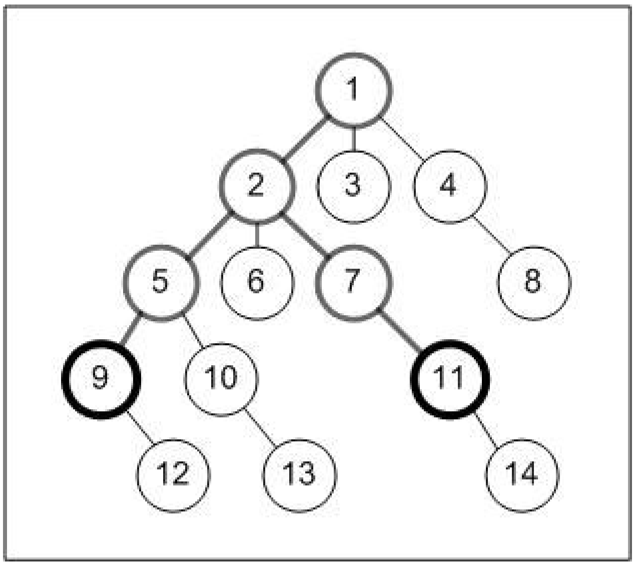

Figure 2 - The two training methods. The CPG from the training set is classified by concepts 9 and 11, and thus is classified also by the ancestor concepts of 9 (9,5,2,1) and 11 (11,7,2,1) up to the root. In the *bottom-up* approach, the CPG vector will be considered twice during the computation of the centroid of concept 2, while in the *top-down* it will be considered only once.

Figure 2 demonstrates an example of a CPG which is classified to concepts 9 and 11, and thus also classified to their ancestors on the path till the root (9,5,2,1) and (11,7,2,1), respectively. When the centroid of concept 2 is created, it will consider the CPG vector twice in the *bottom-up* approach, while in the *top-down* it will consider it only once.

When the concept vector training process is completed, two additional parameters are calculated for each concept node. The first is μ, the *mean* of the similarities of each one of the CPGs classified by the concept (an intermediate or a leaf node) to the concept vector *(the concept's centroid*). Equation 6 shows how the similarity between two given vectors C and d is computed. The similarity is a positive value that represents the angle between the two vectors in the collection terms dimensions. Equation 7 shows how μ (the *mean similarity measure*) is calculated, for a given concept centroid C and a set of n CPGs. The other parameter calculated is σ, which represents the *standard deviation* of the similarities of the classified CPGs vectors, from the concept centroid, as can be seen in Equation 8.

$$Sim(C,d)=\frac{\vec{C}\cdot\vec{d}}{|\vec{C}|\cdot|\vec{d}|}=\frac{\sum_{I=1}^{K}(c_I\cdot d_I)}{\sqrt{\sum_{I=1}^{K}c_I^2\cdot\sum_{I=1}^{K}d_I^2}} \qquad (6)$$

$$\mu(\mathrm{C})=\frac{\sum_{i=1}^{n}Sim(C,di)}{n} \qquad (7)$$

$$\sigma^2(C)=\frac{\sum_{i=1}^{n}(Sim(C,d_i)-\mu(C))^2}{n} \qquad (8)$$

Eventually, we obtain a computed predefined hierarchical conceptual structure, in which each one of the concepts is represented by a centroid, the concept's classified CPGs, the similarities average μ, and the similarities standard deviation σ. These parameters are then used in the classification process.

4.4. The Classification Method

The classification task is to classify a new CPG by one or more concepts (actually, concept *paths*) within the predefined hierarchical concept structure. In our model, we decompose the problem into several simpler problems, using the hierarchical structure and moving from the root of the hierarchy downwards, until a stop criterion holds true. (If the stop criterion is never satisfied along a full path, the classification stops at a leaf node). At each concept in the concept tree, starting from the root, a classification decision is taken. The decision may include one or more sub-nodes, depending on the similarity between the new CPG vector and each concept vector. The similarity of the new CPG is calculated in reference to the concept centroid as shown in Equation 6. The decision whether to classify the new CPG to a concept depends on whether the similarity was higher than the concept *Threshold.* Each concept has a threshold that defines whether a vector, representing a new CPG, belongs to this concept, based on the similarity measure (Equation 6). Equation 9 shows the definition of a *Concept Threshold*, in which μ represents the mean similarity in the specific concept population and σ represents the standard deviation of the similarity in the said population. σ is multiplied in a constant k, defining the Threshold level. The bigger the value of k, the lower becomes the Threshold, which means more, not necessarily relevant, CPGs will be classified, which may increase the error rate.

$$Threshold(C) = \mu - k\sigma \qquad (9)$$

At each concept C a test is made whether the new CPG $_j$ (with a respective CPG vector d_j,) should be classified by it or not as defined in Equation 10.

```
Classify_By( d, C)
{
        If $Sim(d, C) \geq Threshold(C)$                (10)
                Then True;
        Else False;
}
```

Initially, the similarity of the new CPG, represented by a vector of terms, is computed relative to the centroid of the concept tree's root. If the classification test criterion is fulfilled, we examine each of the current concept node's sub-concepts' tree, and the same classification operation will be performed recursively, thus drilling down from the root to the leaves. The classification process might stop at a leaf node or at an intermediate sub-concept of the hierarchy, in the case that no one of the sub-concepts tests resulted in a negative test.

When the test does not evaluate as True in a specific concept (node), we consider one of two approaches. The *conservative* policy is *BestFit*, where the node along the path from the root that has the highest similarity value is chosen as the classified node. The *aggressive* policy is *ParentFit*, in which the parent of the node that just evaluated as False (i.e., the very last node that was still evaluated as True relative to the threshold) is chosen as a classification. Using the *ParentFit* strategy, the classification process is greedy and will drill down in the concepts hierarchy till the test fails. Using the *BestFit* strategy, the process stops at the node in which the highest similarity value was achieved from the root. At the end of the classification process, the new CPG will have one or more classifications which can be described as paths in the concept tree.

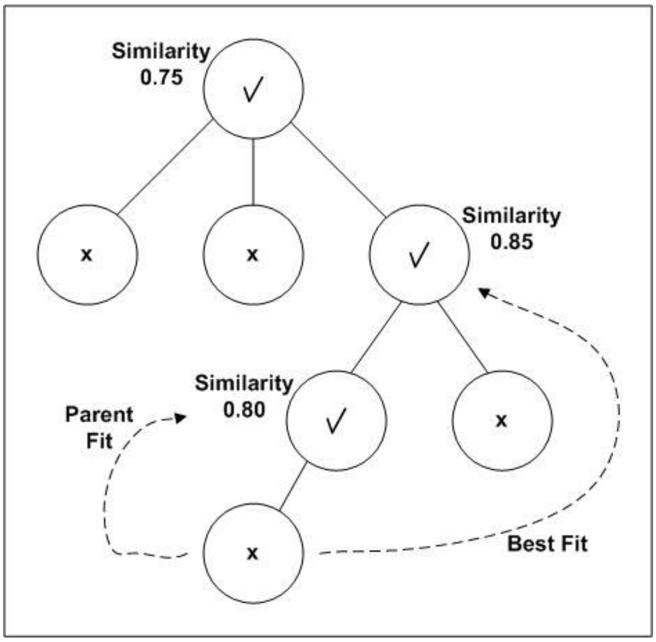

Figure 3 - The two approaches used for the hierarchical-search stop criterion. The Parent Fit stop criterion classifies the new CPG to the last concept node in which the test was true, while the Best Fit stop criterion classifies the CPG to the concept in which the value of the similarity measure was the highest throughout the full classification path.

4.5. The Evaluation Methods

4.5.1. Performance Measures for Hierarchical Classification of Text Documents

Various performance measures exist for text classification, covering different aspects of this task. This section covers the most used performance measures, their benefits and drawbacks.

4.5.2. Basic Precision and Recall Measures

Performance within text classification is usually given in terms of precision and recall, as in information retrieval. A classifier makes a binary decision for each pair <d_j, C_i>. Each decision has four possible outcomes with respect to the correct classification, and can be outlined using a contingency table. Let TP_i be the set of documents that were *classified correctly* to C_i, FP_i be the set of that were *wrongly classified* to C_i, FN_i be the set of documents that were *wrongly rejected*, and TN_i be the set of documents correctly rejected. Pr_i is the *Precision* and Re_i is the *Recall* of a given class *i*, calculated as shown in equation 11.

$$\begin{aligned} Pr\,i &= \frac{|TPi|}{|TPi| + |FPi|} \\ Re\,i &= \frac{|TPi|}{|TPi| + |FNi|} \end{aligned} \qquad (11)$$

Precision and recall as defined in Equation (11) are *class-related* measurements, but the same measures can be computed from the *document's* point of view. Instead of checking how many documents were correctly assigned to each class, we are checking how many classes were correctly identified as relevant to the document.

4.5.3. Hierarchical Precision and Recall Measures

The hierarchical measures are an extension of the traditional precision and recall to the hierarchical classification task. The suggested new measures, as far as known to the authors, consider the set of classifications of a single item (document) as a sub-tree, thus including the whole path in the hierarchy, instead of a set of intermediate concepts, which are the leaves of the said sub-trees. Figure 4 demonstrates an example, in which there are two sub-trees of a CPG: the actual classifications and the classified classifications. While the basic measures consider only the exact concepts, the hierarchical measures consider the full path of the exact concepts, thus measuring whether part of the path was classified correctly.

In order to define the hierarchical measures we define two sub-trees, the actual classifications paths and classified paths, which appear in Figure 4 as sub-trees (A) and (C), respectively. The intersection between these two sub-trees, representing their similarity, appears in

Figure 4 as sub-tree (B). Thus the hierarchical precision is the ratio between sub-tree intersection (B) and the actual classifications paths (A). The hierarchical recall is the ratio between the intersection (B) and classified classifications paths (C).

4.5.4. Coverage

An alternative measure of recall and precision is *coverage*. In this measure, similarly to the hierarchical measures, we represent the real multiple classifications of a document in the concepts hierarchy as a sub-tree, representing the real classifications paths. The resulted multiple classifications, given by the classifier, are represented as another sub-tree. The *coverage* measures the similarity between these two sub-trees, which optimally should be the same. The coverage measures the ratio between the intersection of the sub-trees *RealC*, representing the real (actual) multiple classifications of a CPG, and *Class*, representing the automatically classified paths of a CPG, and the unification of them, as can be seen in Equation 12.

$$Coverage(d) = \frac{\left|\mathrm{Re}\,alC(d)\right| \bigcap \left|Class(d)\right|}{\left|\mathrm{Re}\,alC(d)\right| \bigcup \left|Class(d)\right|} \qquad (12)$$

We originally created this measure to assess the expected accuracy of the concept-based search in Vaidurya. It seems reasonable to use (and even prefer) hierarchical measures when the goal is not necessarily to pinpoint a particular set of concepts, but rather to discover a set of classification *paths* that are as close as possible to the actual set by which the guideline is, or should be, indexed.

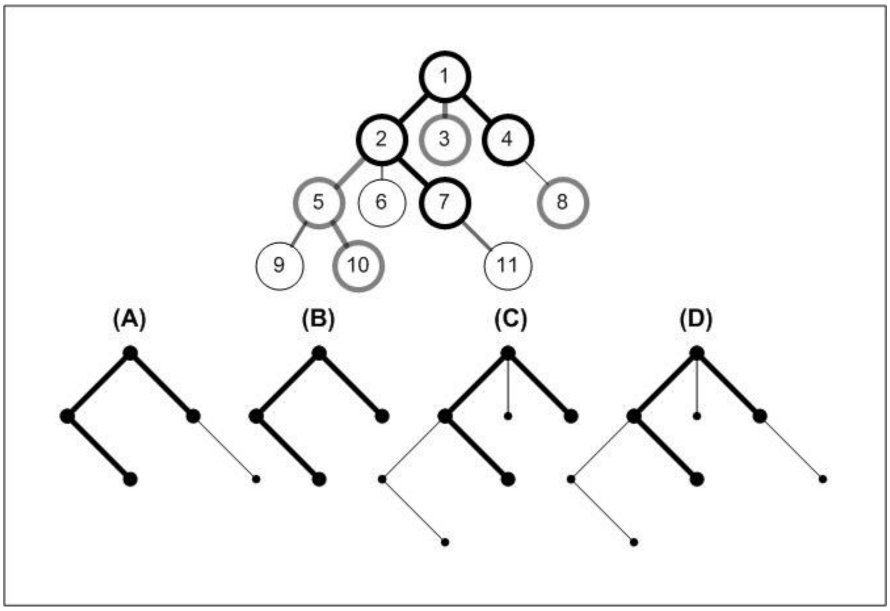

Figure 4 – The tree at the top represents the actual and computed classifications of a CPG in the concepts hierarchy. The dark and bold concepts are the intersection of the actual and classified paths. The grey and bold concepts are either parts of the actual path that were not found or classified, or classified concepts that were wrongly added to the path. The four trees represent different parts of the main tree above. (A) represents the actual classifications, (B) represents the intersection between the actual and classified classifications (appears also at the other trees as bold), (C) represents the resultant classifications, and (D) represents the unification of A and C. While the Basic measures refer only to concepts 4 and 7, which are the leaves of the actual classification tree (A), the hierarchical measures and the coverage refer to the full paths. Thus, the hierarchical precision is |B|/|A|, the hierarchical recall is |B|/|C| and the coverage is |B|/|D|. (|*X*| *refers to the number of concepts within the tree X*)

4.5.5. Micro averaging

The Precision and Recall of the entire document test collection are averaged. This average indicates the accuracy of the classification from each document's point of view, which means how many concept classes were classified correctly to each guideline document.

4.5.6. Macro averaging

Precision and recall are evaluated for each *concept class* and then averaged; this provides us with the accuracy of the classification from the *class* or *concept* point of view. Thus, the evaluation of the results of a multiple hierarchical classification process can use either basic or hierarchical measures, and, orthogonally, either micro or macro averaging.

4.6. The Evaluation Process

We performed an evaluation based on a set of CPGs classified along two hierarchical concept structures. In this evaluation we wanted to evaluate the different settings of the classification algorithm. First, we examined several methods of selecting the optimal text-based features (terms) for representing the guideline documents, eventually settling on selection of terms using the *HMI* measure; this process was relatively brief, since it was not the main focus of the current study. We then considered in more depth the main topic of this study, the hierarchical training and classification methods and several evaluation methods to assess their impact. The measures we used to evaluate the different classification settings were: *Basic Micro Recall, Basic Micro Precision, Hierarchical Micro Recall, Hierarchical Micro Precision* and *Coverage*.

4.7. The Evaluation Data Set

The DeGeL Library is an electronic repository and a set of CPG specification, retrieval, and application tools, but at this point in time it is not an extensive medical content repository. We have used the NGC CPGs collection and uploaded it into the DEGEL server just for the experiment. The NGC website is a repository of CPGs classified along the MeSH concepts. The CPGs are classified along two tree-like hierarchical concepts, Disorders and Therapies.

Each concept tree has roughly 2,500 unique concepts in a tree-like structure; overall, there are 5,407 concepts. In several regions, the concept trees are 10 levels deep, but the average depth is around 4-6 levels. There are 1,136 CPGs; each CPG may have multiple classifications indexed by both of the concept trees, including indices that belong to the same tree but at different levels (nodes), not necessarily leaves. Overall, there exist 11,360 guideline-concept classifications. Thus, CPGs have on average 10 classifications per guideline; however, there are CPGs that are classified by 90 concepts.

Since we wanted to use a repository of existing, classified guidelines, and not to classify them again, the evaluation uses only the current NGC axes (that is, we did not add any new indices based on the DEGEL axes). The NGC guidelines were uploaded into the DEGEL framework and the two NGC axes, Disorders and Therapies, replaced, only for this experiment, the usual DEGEL axes.

Since we wanted to have a training set and a test set, but the axes structure was hierarchical and the ratio of instances to concepts was low, we could not select the training-set instances randomly, unlike the typical situation when performing classification using a "flat" set of classes. We wanted to have a uniform coverage of the training set within the taxonomy. Note that while the taxonomies include roughly 2,500 concepts each, we had only 1,136 documents in the entire collection. In order to overcome this low rate of documents per concept, we exploited the multiple classifications of each document. We had a stopping criterion which defined the minimal amount of directly classified CPGs that need to be classified by any concept (e.g., having at least a single guideline classified by each concept). Note that at the end of the process, we had at least *x*, which was set in the training phase, CPGs classified (or presented) by each concept, while for several concepts, we had more than one example, since typically each CPG is classified by more than one concept. When considering each concept's ancestors as implicitly classifying the CPGs that are classified by that concept, we achieved a higher rate of documents per concept. The CPGs which were not selected in the training set were the testing set. Several experiments were performed. The training set in our experiments consisted of, on average, 640 guidelines, while the testing set consisted of, on average, 460 guidelines; the rest were CPGs with no classifications which we exclude from the evaluation.

Since many of the concept nodes served as indices for up to only two guidelines, we experimented with the option of trimming the concept trees so as to aggregate several of the leaves and low intermediate concepts together, by "folding" them into the higher level classes to which they belong, thus increasing the ratio of guidelines to concepts. (E.g., "fold" a concept such as tuberculosis into the higher level concept of an infectious pulmonary disease). Thus, the resulting folded trees included the first four levels of the hierarchies, which still included a significant number (1,938) of the concepts. Most of the experiments we report were performed on that "folded" set.

5. The Evaluation Results

5.1. Determining the Feature Selection Method

We applied each FS measure, described in detail in the Feature Selection section, within a range of settings. We examined several feature selection methods, in particular, Words distribution (WD2Idf) and Mutual Information (MI) measures. Each FS threshold has a different effect, based on the measure definition, while the effect of the threshold might be considered as negligible in the MI measures; this is not the case with the Word Distribution measure.

Table 1 summarizes the results of 60 experiments in which different FS measures were tested. Table 1 demonstrates the performance achieved for each feature selection method. Best precision (hierarchical and basic) was achieved for WD2Idf at a threshold of top 40% and bottom 1%. As can be seen from the table the other method performed similarly, while on average the HMI method performed best. The best Recall (Hierarchical and Basic) performance was achieved for the HMI method at a threshold of top 10%. On average, the HMI outperformed the WD2. Based on the results, it seems that when the threshold was reduced the recall increased, but the precision decreased when the HMI was used. With the WD2, changing the threshold changed both Precision and Recall by the same magnitude. According to the coverage measure, the best results were with the HMI at top 50%.

5.2. Hierarchical Mutual Information versus Mutual Information

One of our objectives within the Feature Selection experiments was to compare the Hierarchy Mutual Information and the regular Mutual Information. We wanted to learn whether considering the structure of the hierarchy in the MI method, thus using HMI, would result in a different performance. Note that in the HMI each class (concept) includes not only the documents classified directly but also the ones classified to their descendents.

Table 2 shows the performance achieved for both of the techniques in 36 experiments using a similar mixture of training methods and stop criteria. As shown in the table, and contrary to our expectations, we did not observe any substantial difference between the two methods. There is a slight advantage to the MI measure in the Precision measures, while there is a slight advantage to the HMI in the Recall measures and the coverage.

Table 1 - The performance results for each feature-selection method

Method	Trim Level	Basic Micro Precision	Basic Micro Recall	Hierarchical Micro Precision	Hierarchical Micro Recall	Coverage
HMI	Top10%	16.83%	53.84%	46.26%	66.18%	34.75%
	Top20%	17.15%	49.14%	48.13%	61.89%	34.31%
	Top50%	17.50%	49.80%	49.76%	62.98%	35.05%
Average HMI		17.16%	50.93%	48.05%	63.68%	34.70%
Wd2Idf	Up 40% Down 1%	17.66%	47.45%	50.09%	60.17%	33.80%
	Up 40% Down 10%	15.87%	37.17%	47.92%	49.37%	26.52%
	Top10% up & down	15.64%	34.97%	49.10%	47.81%	26.87%
	Top20% up & down	14.41%	33.55%	44.46%	45.39%	22.05%
Average Wd2Idf		15.89%	38.29%	47.89%	50.69%	27.31%

Table 2 - The performance results for the MI and HMI feature selection measures

Measure	HMI	MI
Basic Micro Precision	19.02%	19.29%
Basic Micro Recall	45.41%	44.21%
Hierarchical Micro Precision	52.41%	52.72%
Hierarchical Micro Recall	55.56%	53.73%
Coverage Average	29.99%	28.90%

5.3. Training Methods

We compared two training methods: Top-Down and Bottom-Up, based on 400 experiments including various features selection methods and classification methods and different folding levels, shown in Table 3.

Table 3 - The performance results of the training methods[1]

Measure	Top Down	Bottom Up	P
Basic Micro Precision	12.98 ± 01%	8.58 ± 01%	< 0.01
Basic Micro Recall	23.43 ± 07%	20.72 ± 08%	0.11
Hierarchical Micro Precision	55.76 ± 04%	51.73 ± 08%	0.22
Hierarchical Micro Recall	29.27 ± 09%	25.63 ± 09%	0.09
Coverage	13.05 ± 02%	10.40 ± 01%	0.01

[1] The *Top-Down* training approach, in which each CPG is considered only once even if it is classified by multiple sub-concepts of a specific concept, outperforms the *Bottom-Up* training approach, in which a concept vector is computed based on each CPG vector even if appearing more than once within its sub-concepts.

As expected, the hierarchical measures are higher, since they do not require a precise match of a set of particular concepts, but rather, a suggestion of a large portion of the correct classification path(s). The Top Down training algorithm consistently achieved a higher score for all of the measures, while in part of them a significant increase was achieved.

5.4. Stop-Criteria Methods

In this investigation, we compared two classification methods: Parent-fit and Best-fit, which were described earlier. Table 4 shows the performance observed based on 600 experiments with an averaged equal use of various feature selection settings, and different folding levels and classification thresholds. Better results were achieved when the conservative Best-Fit approach was used, compared to the aggressive Parent-Fit one. While all of the measures show a distinct advantage to the Best-Fit over the Parent-Fit, the differences are greater for precision and coverage.

Table 4 – The performance results for the stopping criteria methods[2]

Measure	Best Fit	Parent Fit	P
Basic Micro Precision	19.24 ± 09 %	15.50 ± 08 %	0.01
Basic Micro Recall	37.76 ± 29 %	36.72 ± 35 %	0.43
Hierarchical Micro Precision	56.40 ± 20 %	48.28 ± 22 %	0.02
Hierarchical Micro Recall	43.25 ± 30 %	38.65 ± 32 %	0.21
Average Coverage	28.82 ± 16 %	14.05 ± 11 %	< 0.01

[2] The conservative *Best-Fit* approach, in which the concept for which the highest similarity was achieved, within the path from the root, is chosen, outperformed the aggressive approach *Parent-Fit,* in which the last concept that passed the test is chosen.

5.5. The Influence of the Classification Threshold

As explained earlier, for each concept in the hierarchy there is a standard deviation of the documents in the concept relative to the centroid. The threshold for determining whether or not a new document belongs to the concept is based on the standard deviation of similarities between the concept's documents, where the size of the threshold is a multiplier (*k*) of this standard deviation (Eq. 9). Since *k parameter* can influence the performance in different ways within the classification approaches (ParentFit/BestFit), we examined the *k parameter* settings for each approach separately. Table 5 demonstrates the performance achieved over a total of 100 experiments with the BestFit approach, where the range of *k* is [-0.2, 0.2].

Table 5 – The performance results for different settings of *k* at the BestFit approach[3]

	Num of Standard Deviations				
Measure	**-0.2**	**-0.1**	**0**	**0.1**	**0.2**
Basic Micro Precision	5.82%	11.55%	14.15%	**16.62%**	14.68%
Basic Micro Recall	0.22%	1.65%	8.22%	29.41%	**52.68%**
Hierarchical Micro Precision	**67.22%**	66.12%	57.87%	54.56%	39.07%
Hierarchical Micro Recall	0.68%	3.42%	12.92%	42.53%	**66.50%**
Average Coverage	0.59%	2.53%	8.90%	26.98%	**31.99%**

The results in Table 5 demonstrate that while the best Basic Precision was achieved with a *k* of 0.1 standard deviation, the best Hierarchical Precision was achieved with a negative setting

[3] The value of *k* has a tremendous influence on the classifier performance. While most of the measures outperformed for *k*=0.2, the highest hierarchical micro precision was achieved for *k*=-0.2.

of -0.2. The highest Recall results (both Hierarchical and Basic) and Coverage were achieved by using a *k* of 0.2. The best Basic Precision was achieved with *k* = 0.1.

We examined the results also for the ParentFit approach. Table 6 demonstrates the results, based on 100 experiments.

Table 6 – The performance results for different settings of *k* at the ParentFit approach

	Num Of Standard Deviations				
Measure	**-0.2**	**-0.1**	**0**	**0.1**	**0.2**
Basic Micro Precision	5.60%	10.55%	13.41%	11.83%	4.41%
Basic Micro Recall	0.22%	1.79%	9.38%	37.82%	84.46%
Hierarchical Micro Precision	67.18%	61.85%	54.79%	37.92%	10.79%
Hierarchical Micro Recall	0.70%	3.63%	14.15%	50.77%	81.15%
Average Coverage	0.60%	2.56%	8.89%	22.14%	10.06%

The reasonable range of *k* value is [0, 0.1] where the Precision is still high enough; below and above this range the Precision is very low. Yet, examining the Recall rate, the recommended range is [0.1, ∞]. Therefore, we concluded that the optimal value of *k* is 0.1; the Precision and the Recall rate both seem sufficiently useful for the purposes of tasks such as ours (support of semi-manual CPG classification).

5.6. The Impact of Tree-folding

Since our experiments were based on folded tree we were concerned that the folding might cause deviation in the classification result. To insure that there is no "noise" caused by the folding, we have compared various levels of folding (folding level means the maximum tree depth remaining after folding). The analysis is based on 600 experiments and includes various settings.

Table 8 - The performance incensement when the ratio of CPGs per concepts increased

	Folding Level		
Measure	**2**	**3**	**4**
Basic Micro Precision	17.54%	9.70%	6.31%
Basic Micro Recall	37.48%	29.34%	22.95%
Hierarchical Micro Precision	51.93%	45.07%	44.37%
Hierarchical Micro Recall	41.64%	35.72%	31.84%
Average Coverage	19.08%	13.71%	11.44%
# Of Categories	87	556	1938
Average # of classified docs. Per Category	115	29	12

Table 8 shows that there is a correlation between the folding level and the classification performance. As we expected, the performance increased as the folding level decreased. The interesting fact is that while the flat (basic) measures significantly decreased as the folding level increased, the hierarchical measures showed a considerably lower correlation, and therefore the decrease in them was less sharp. This emphasizes the fact that the hierarchical measurements are less dependent on the folding level and are more effective in understanding the hierarchical classification abilities.

6. Discussion

Classification of CPGs in clinical medicine is important for multiple clinical tasks. We have previously suggested and presented a complete framework (DEGEL) that manages the guideline specification, classification, storage, search, retrieval, and application process. A unique aspect of the domain is the necessity for a multiple-classification approach using a hierarchical classification structure. In the current study, we presented a full process for supporting the semi-automated task of classification of a new guideline by a medical expert, who appreciates a suggestion for several potential paths in the guideline classification hierarchies.

There is a marked influence exerted by the intended user and guideline-related task on the relevant evaluation measure (and thus, on the policy chosen during classification) that would be considered optimal for a guideline-classification tool.

In this study, we used several evaluation measures. We used the basic Precision and Recall measures, commonly used in Text Categorization, which assess the accuracy of the classification while ignoring the hierarchical relation between the concepts (classes). As can be seen from the results these two measures are very challenging when used as is, especially in the particular test-set in which there are roughly 2,000 concepts (classes). In addition, we suggested the hierarchical Precision and Recall, which are based on the original Precision and Recall but consider the relation between the concepts, based on the hierarchical structure. We propose also the coverage measure, which combines the hierarchical Precision and Recall measures. The hierarchical measures (including the coverage) assess the classification accuracy considering the hierarchical structure properties, such as that CPGs classified to lower concepts are implicitly classified to their ancestors as well. The hierarchical measures are useful for assessing the classification, in terms of the expected accuracy for the use of a concept-based search such as is, for example, implemented in the Vaidurya search engine. Since the search in Vaidurya for a specified concept includes all the CPGs classified by the concept and by its sub-concepts recursively, the hierarchical measures may indicate the expected retrieval accuracy. Such a requirement is different from the requirements of a application in which each CPG must be classified accurately by exactly the one right concept. A hierarchical classification evaluation measure also better captures the intended support of a human editor who is presented with a proposed set of classification paths, and who can then slightly modify a path by either extending it to a lower (more specific) level or pruning a path that seems too specific. That type of support was one of the motivations for our research. For example in Table 5 we can see that the *basic micro precision* value is quite steady at all the settings of the *k* parameter while at the hierarchical measures there is a clear change in performance. This property is very important especially in the context of a concept based search engine if we would like to use automatic hierachical classification (without any human editing).

The two types of measures discussed here (basic and hierarchical) and the considerations regarding them are especially relevant to choosing the appropriate settings of the classifier for each task. These settings depend on that measure of success to be used. When we want an exact classification, in which each CPG must be classified by the proper concept, the basic Precision and Recall measures should be preferred, but when the purpose is concept-based search, as in Vaidurya (whether for assisting a human classifier or as a final standalone classification) the hierarchical measures should be considered.

The focus of this study was *not* on the feature-selection measures, but rather on the *hierarchical classification* and *evaluation* methods. However, we did look at several different options. Eventually, we used two feature (term) selection methods to represent the text-based guidelines, since in preliminary experiments they seemed at least as good as any of the other methods: The HMI measure and the W2Idf measure. The results in Table 1 indicate that the best results, on average, were achieved when HMI used.

In general, we conclude that in our particular setting, the HMI measure proved to be a more effective feature-selection method compared to the MI and W2IDF measures for feature selection. Table 2 shows several of our experiments using the MI, where no significant improvement was observed. We believe that if the rate of CPGs per concept were higher than it is in the current test set (which is approximately 1300/1938) we could see a difference in the results.

We suggested and evaluated two training methods *Top-Down* and *Bottom-Up*, based on the hierarchical structure; in all the measures we observed that the *Top-Down* outperformed the *Bottom-Up*.

As shown in Formula 4, the *x* parameter represents the relative contribution of descendent concepts. In all of the described experiments, the value of *x* was kept as 0.5, since our initial experiments did not show any improvement when its value was other than 0.5. However, it is quite possible to envision giving *x* a value between 0 and 0.5, thus reducing the effect of descendent concepts, perhaps in other domains or different data sets. It is, in theory, also possible to assign to it a value higher than one, to increase specificity to a particular sub-concept.

In the classification phase, we suggested two policies: *Parent-Fit*, which is more greedy and aggressive, and *Best-Fit*, which is more conservative. The results had shown that the *Best-Fit* outperformed the *Parent-Fit* for all measures.

When we examined the optimal number of standard deviations needed for the best classification performance, we saw that the Precision was higher when a lower number of standard deviations was set, while a higher Recall was achieved for a higher number of standard deviations. This is quite intuitive, since when a higher number of standard deviations is used, the threshold at each concept becomes lower, and thus more CPGs are classified to varying concepts, not necessarily correctly, resulting in a higher Recall (although Precision usually decreases); the opposite, namely reducing the number of standard deviations allowed (thus increasing the threshold for classification by the concept), results in higher Precision, although a smaller number of guidelines are correctly classified to the concept (a measure equivalent to Recall).

7. Summary and Conclusions

In this paper we have proposed an approach for multiple classifications of CPGs using a given conceptual hierarchy. We evaluated the suggested approach on a test set of CPGs. Based on the evaluation, we can conclude that the best performance is achieved when the Mutual Information feature selection technique was applied. The best training technique proved in this case to be *Top-Down* motion through the hierarchy, and the best classification method for the evaluation configuration parameters we used was *Best-Fit*.

Eventually, the classifier settings should be set based on the requirements of the resultant classification. A conservative approach, such as *Best-Fit*, in which the concept with the highest similarity along the classification path is selected by the classification process, is optimal for reducing the risk of incorrect classification, while still supporting the recursive search and retrieval performed by Vaidurya for clinical end-users. However, a risky aggressive approach, such as *Parent-fit*, in which the furthest node along the classification that passed the threshold is chosen, might be better for a medical expert or knowledge engineer working with the IndexiGuide classification tool; the expert needs a suggestion for the most specific classification(s) possible, and can always correct a wrong suggestion by going up a level if needed.

To summarize, the task of hierarchical multiple classification is a very challenging one, and involves application of theories originating from the areas of text classification and decision theory, especially when viewing the given semantic-classification hierarchy as a predefined decision tree. Further investigation of multiple-classification approaches should be carried out, in which both the structure of the hierarchy and the classification-decisions pattern should be considered.

8. Acknowledgements

This research was supported in part by NIH award LM-06806. We thank the undergraduate students who did initial prior studies which lead eventually to this study: Arale Yehezkel and Revital Levy, who worked on a Naïve Bayes and KNN classifier version, and Nir Regev, Rami Umschweif and Zeev Nahum, who applied a clustering technique for the classification of CPGs hierarchically.

9. References

[1] F. Sebastiani, Machine learning in automated text categorization, *ACM Computing Surveys* (2002) 34(1):1-47.

[2] J.M Grimshaw and I.T. Russel, Effect of clinical guidelines on medical practice: A systematic review of rigorous evaluations, *Lancet* (1993) 342: 1317–1322.

[3] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. A. Greenes, R. Hall, P. D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. H. Shortliffe & M. Stefanelli. Comparing computer-interpretable guideline models: a case-study approach, *Journal of American Medical Informatics Association*, Vol. 10, No. 1 (2003), pp. 52-68.

[4] Y. Shahar, O. Young E. Shalom, M. Galperin, M. Mayaffit, R. Moskovitch, and A. Hessing, A framework for a distributed, hybrid, multiple-ontology clinical-guideline library and automated guideline-support tools, *Journal of Biomedical Informatics* (2004) 37, 25-344.

[5] Medical Subject Headings (MeSH): http://www.nlm.nih.gov/mesh/meshhome.html (Accessed: 20 March 2006)

[6] B.L. Humphreys and D.A. Lindberg, The UMLS project: making the conceptual connection between users and the information they need, *Bull Med Libr Assoc*. (1993) 81(2): 170-177.

[7] R. Moskovitch, A. Hessing and Y. Shahar, Vaidurya – A concept-based, context-sensitive search engine for clinical guidelines, *Proceedings of the 2004 International Medical Informatics Conference: MedInfo'04*, San Francisco, CA, USA (2004).

[8] (343/2005) R. Moskovitch, A concept-based, context-sensitive, multiple-ontology search engine for clinical guidelines, Msc Thesis, Ben Gurion University, Israel (2006).

[9] N.S. Fuhr, Hartmanna, G. Lustig, M. Schwantner and K. Tzeras, Air/X – A rule-based multi-stage indexing system for large subject fields, *Proceedings of the international conference of Recherche d'Information Assistée par Ordinateur* (1991) 606-623.

[10] T. Joachims, Text categorization with support vector machines: Learning with many relevant features. *Proceedings of the European Conference on Machine Learning '98* (1998).

[11] Y. Yang, Expert network: Effective and efficient learning from human decisions in text categorization and retrieval, *Proceedings of the Annual International conference of the ACM Special Interest Group on Information Retrieval* (1994)*,* 13-22.

[12] D.D. Lewis and M.A. Ringuette, A comparison of two learning algorithms for text categorization, *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval* (1994) 81-93.

[13] A.A. Weigend, E.E. Wiener, and J.O. Pedersen, Exploiting hierarchy in text categorization, *Information Retrieval* (1999) 1(3), 193-216.

[14] S. T. Dumais and H. Chen Hierarchical classification of web content, *Proceedings of the Annual International conference of the ACM Special Interest Group on Information Retrieval*, (2000) pp. 256-263.

[15] A. Sun and E.-P. Lim. Hierarchical text classification and evaluation. *Proceedings of the IEEE International Conference on Data Mining* (2001) 521-528.

[16] W.R. Hersh and R.A. Greenes, SAPHIRE – an information retrieval system featuring concept matching, automatic indexing, probabilistic retrieval and hierarchical relationships, *Computers and Biomedical Research* (1989) 3, 410-25.

[17] W. Pratt and L. Fagan, The usefulness of dynamically categorizing search results, *Journal of American Medical Informatics Association* (2000), 7(6), 605-617.

[18] D. Koller and M. Sahami, Hierarchically classifying documents using very few words, *Proceedings of the Fourteenth International Conference on Machine Learning,* (1997) 170-178.

[19] M.E. Ruiz and P. Srinivasan, Hierarchical neural networks for text categorization, *Proceedings of the Annual International conference of the ACM Special Interest Group on Information Retrieval*, (1999) 281-282.

[20] G. Salton, A. Wong, and C.S. Yang, A vector space model for automatic indexing. *Communications of the ACM* (1975), 18:613 620.

[21] J. J. Rocchio, Relevance feedback in information retrieval. In: Gerard Salton, ed., *The SMART retrieval system: experiments in automatic document processing* (Prentice-Hall, Engelwood Clis, US, 1971), 313-323.

[22] M.F. Porter, M.F, An algorithm for suffix stripping, *Program* (1980) 14(3) :130-137.