Domain Ontology Guided Feature-Selection for Document Categorization

Bill B. Wang, R I. (Bob) McKay, Hussein A. Abbass, Michael Barlow

School of Computer Science, University College, ADFA, University of New South Wales
Canberra, ACT 2600
Email: {biaowang, rim, abbass, spike}@cs.adfa.edu.au

Abstract: We present a novel method employing a hierarchical domain ontology structure to select features representing documents. All raw words in the training documents are mapped to concepts in a domain ontology. Based on these concepts, a concept hierarchy is established for the training document space, using is-a relationships defined in the domain ontology. An optimum concept set may be obtained by searching the concept hierarchy with an appropriate heuristic function. This may be used as the feature space to represent the training dataset. The proposed method aims to solve some drawbacks suffered by text classification algorithms and feature selection algorithms. One major difficulty for text classification algorithms, especially for machine learning approaches, is the high dimensionality of the feature space. The second major difficulty is to obtain a training dataset of good quality, which is crucial to the performances of almost all text classifiers. Experimental results show that our method solves these problems more reasonably and more effectively than existing methods.

1. Introduction

Automatic text categorization (classification) [1] is the task of assigning natural language texts to one or more pre-defined categories based on their content. The emergence of the Web has led to an exponential increase in the volume of on-line documents. Text categorization is increasingly important to help people find information from these vast resources. At the same time, text categorization presents huge challenges arising from specific characteristics of the learning task. These include the large number of features, feature dependency, multi-modality and large training sets.

A growing number of statistical learning methods have been applied to this problem in recent years, including Bayes belief networks [2, 3], decision trees [2], support vector machines [4, 5], neural networks [6] and K-nearest neighbor (KNN) classifiers [7, 8]. However, these methods all face a key difficulty, the high dimensionality of the feature space. The feature space for a text classifier consists of the unique terms – which may be words, phrases or strings of characters – that occur in training documents. It may contain tens or hundreds of thousands of terms for even a moderate-sized text collection. This is prohibitively high for many classification algorithms, especially for some learning algorithms [9].

Many researchers have shown that similarity based categorization algorithms, such as KNN and centroid based classification, are very effective for large document collections [11]. A cross-experiment comparison [10] between 14 major categorization methods, including KNN, decision tree, naïve Bayes, linear least squares fit, neural network, SWAP-1, Rocchio, etc., has shown that KNN is one of the top performers, and it performs well in scaling up to very large and noisy categorization problems. However, these effective categorization algorithms still suffer disadvantages from high dimensionality that greatly limit their practical performance.

Empirical and mathematical analysis [15, 16] has shown that finding the nearest neighbors in high-dimensional space is very difficult because most points in high-dimensional space are almost equi-distant from all the other points. In fact, in many document data sets, only a relatively small number of the total features may be useful in categorizing documents, and using all the features may adversely affect performance. So determining how to reduce the length of document vectors effectively and reasonably is a challenge for categorization researchers. Stop words lists [26] and word stemming [27] are some of the earliest effort in this problem. In recent years, many term-weighting and feature-selection algorithms [2, 9, 11, 12, 13, 14] have been developed, to reduce the feature space without sacrificing categorization accuracy. However, the effectiveness of these algorithms heavily depends on the quality of training dataset. This is a major drawback for text classification methods, as the creation of high quality datasets may be very expensive.

The performance of both the text classification algorithms discussed above, and of feature selection algorithms, depend on the quality of training dataset. The KNN classifier is an instance-based classifier, which means a training dataset of high quality is particularly important. An ideal training document set for each particular category will cover all the important terms, and their possible distribution in this category. Otherwise, a text that uses only some key words out of a training set may be assigned to the wrong category. In practice, however, establishing such a training set is usually infeasible.

This paper instead uses a concept set as the feature space to represent the documents. We present a novel method for finding an appropriate concept set. It searches a domain-specific concept hierarchy to find a good concept set for a KNN text classifier. The goal of this approach is to overcome the above drawbacks.
effectively and reasonably. Obviously, different pre-defined categories have their own taxonomic standards. This means they use different concept levels in a domain-specific concept hierarchy to identify document content. For example, the names of different kinds of heart disease are key index terms to identify text content if each category represents a type of heart disease. However, if the desired categories are the 23 major MeSH (Medical Subject Headings) disease categories, all these names may be mapped to the concept "heart disease" without any negative influence on the performance of the classifier. The principal idea of our approach is to establish a hierarchical concept structure based on a domain-specific concept ontology appropriate to a particular training document set, and then to search this concept hierarchy to find an optimum concept set to represent the documents. As a consequence of the effective reduction of the dimensionality of the vector space, using higher level concepts to represent documents may also be expected to improve the ability of the categorization to generalize to new documents. Furthermore, when new terms (e.g. names of new medicines or of newly emergent viruses) arise in related categories, these new terms can be simply added to their proper positions in the concept hierarchy. It won't be necessary, as it is with existing methods, to re-train the classifier.

This paper is structured as follows: Section 2 briefly introduces the notion of domain-specific concept ontology and UMLS knowledge resources, Section 3 describes the process of this system, some experimental results and discussions are presented in Section 4, finally the conclusion is given in Section 5.

2. Domain-Specific Concept Ontology and UMLS Knowledge Resources

The term ontology has various meanings when it is used in different ways and in different disciplines. However, computer scientists use the term ontology to describe formal descriptions of objects in the world, the properties of those objects, and the relationships among them. In artificial intelligence, according to Gruber [17], an ontology is a specification of a conceptualization. It defines the vocabulary of a domain, and constraints on the use of terms in the vocabulary.

In our research, a term is a sequence of alpha-numeric characters which is delimited by white space or punctuation marks. A domain-specific concept ontology specifies the concepts that are used to represent documents. A concept represents a unit of meaningful information in this domain. A concept may consist of one or more terms. A domain ontology also specifies the categories attached to these concepts, and the relations (ISA in this paper) which exist between concepts and categories (Figure 1). The hierarchical concept structure, which we use for a particular training document set, is a part of a domain-specific concept ontology based on terms used in the training set. The process to establish this structure is introduced in Section 3.

The Unified Medical Language System (UMLS) [19], a set of knowledge sources developed by the US National Library of Medicine, can be viewed as a complete concept ontology for medical domains. It consists of three sections: a metathesaurus, a semantic network and a specialist lexicon; and contains information about medical terms and their inter-relationships. It is organised by concept, and

![Domain-Specific Concept Ontology](image)

Figure 1. A Sketch of a Domain-Specific Concept Ontology

contains over 800,000 concepts and 1.9 million entries. Various types of relationships between concepts are defined in this system. ISA is the primary relationship. We used this relationship to establish the hierarchical concept structure for a particular training set containing documents in the medical domain.

3. Establishing Concept Representation

There are four major steps to establishing a concept representation for documents.

1. Map raw terms to concepts based on UMLS
2. Establish a concept hierarchy for the training set
3. Search the concept hierarchy to obtain the optimal concept set
4. Establish a new feature model for both training and test documents

3.1 Mapping Raw Terms to Concepts

The most straightforward representation of documents relies on term vectors. The major drawback of this basic approach for document representation is the length of the feature vectors, usually more than 10,000 terms. In the application of text categorization, however, completely different terms may represent the same concepts. In some cases, terms with different concepts can even be replaced with a higher level concept without negative effect on performance of the classifier. For example, ANEMIA and LEUKEMIA can be replaced with the higher level concept HEMATOLOGIC DISEASE, in many situations of text categorization, without loss of the classifiers' accuracy. Obviously,
mapping terms to concepts is an effective and reasonable method to reduce the dimensionality of the vector space.

The mapping process relies on the API provided by the UMLS system. We use the mapping function provided by the UMLS query interface. We aim to find the 'longest concept units' (LCUs) in documents. An LCU is an independent concept defined by a string of continuous terms and any other string of continuous terms, which contains this string, cannot define an independent concept. For example, consider the sentence "AIDS is a kind of human immunodeficiency virus." According to the mapping algorithm defined below, we will get two concepts: 'AIDS' and 'HIV'. 'Human' and 'virus' are not recorded as independent concepts, even though they do occur as independent concepts in the concept ontology. We take this approach because an LCU is usually more meaningful for identifying the content of a document.

**Term2Concept Mapping Algorithm:**

**Input:** a sentence consisting of n terms \( q = \{ t_1, t_2, \ldots, t_n \} \)

```
C = \emptyset
while(\( q \neq \emptyset \))
    \( \Phi_i \leftarrow \emptyset \)
    while(\( \Phi_i \neq \emptyset \))
        c \leftarrow mapping(\( \Phi_i \))
        if c \neq Null
            then put c in C, remove \( \Phi_i \) from \( \Phi \)
            else remove \( t_i \) from \( \Phi \)
    loop
return C as the concept set for this sentence
```

Through this mapping process, we will obtain concept sets for individual documents. Each will include all distinct concepts and their frequency for the individual document. We will also obtain a shared concept set for the document set, which includes all distinct concepts from the whole document set.

### 3.2 Establishing the Concept Hierarchy

The UMLS query interface provides a parent query function for retrieving parents of concepts. The concept hierarchic structure is established by repeatedly querying 'parent' from shared concepts up to the root of the semantic network. The completed concept hierarchic structure is a fully-connected graph rooted at 'top-type'. For instance, suppose that there are only five distinct concepts as below occurring in a document set.

```
[Mastadenovirus, AIDS, Human Immunodeficiency Syndrome, Alfamovirus, Dengue Virus]
```

Based on this shared concept set, we will get the concept hierarchic structure in Figure 2. From this structure, we can see that several combinations of different level concepts can be chosen to represent documents for different taxonomic standards, e.g. [Virus], [DNA Virus, RNA Virus] and [DNA Virus, Retroviridae, Astroviridae]. All original concepts can be mapped to these higher level concepts.

**3.3 Search Concept Hierarchy**

In this paper, we use a hill-climbing algorithm to search the concept hierarchic structure obtained in the previous step to find the optimum representation (a set of concepts) for a particular document set. Our aim is to use a set of concepts to represent training documents which is as high in the concept hierarchy as possible without loss of categorization accuracy.

First, we specify that all concept nodes, except the root node, have an out edge to their parent nodes. Then we establish a copy of the hierarchical structure for each document. We assign the frequency of each concept occurring in the document to the edge leading into the parent concept node. Thus the frequency for the edge leading out of a parent concept node is the sum of the edge frequencies of all child nodes.

**Figure 2. A Sample Concept Hierarchical Structure**
The vital problem is to define an appropriate heuristic function for the hill climbing search algorithm. In this model, each document is considered to be a vector in the concept-space. In its simplest form, each document \( \vec{d} = (c_{f_1}, c_{f_2}, ..., c_{f_n}) \) is represented by the concept-frequency (CF) vector, where \( c_{f_i} \) is the frequency of the \( i \)th concept in the document.

In order to account for documents of different lengths, each frequency is normalized by dividing by the document length.

In the vector-space model, the similarity between document \( i \) and document \( j \) is commonly measured using the cosine function, given by

\[
s_{ij} = \cos(\vec{d}_i, \vec{d}_j) = \frac{\vec{d}_i \cdot \vec{d}_j}{||\vec{d}_i||_2 \cdot ||\vec{d}_j||_2}.
\]

Since the document vectors are of unit length, the above formula simplifies to

\[
s_{ij} = \cos(\vec{d}_i, \vec{d}_j) = \vec{d}_i \cdot \vec{d}_j.
\]

Finally, we define the heuristic function for the hill climbing search algorithm as below.

\[
\text{heuristic} = (1 - \frac{n}{\alpha}) \sum_{i \in D} D_{cl} \mid D_{cl} \subseteq D_{KNN} \beta > 0
\]

\[D \text{ is the set of all documents in the training set. } D_{cl} \text{ is the set of all documents that belong to the same category, which contains document } i, \text{ in a particular document set. } D_{KNN} \text{ is the set of } k \text{ nearest neighbors for document } i \text{ in the training set. } n \text{ is the dimensionality of feature space. } \alpha \text{ is the number of leaf nodes. } \beta \text{ is a constant. The first part of the right side of the equation is a reward factor, intended to encourage the use of high level concepts in the feature set with despite a limited loss of categorization accuracy. We suggest that } \beta \text{ is chosen less than 0.05. This means that the effect on heuristic value resulting from the reward is at most 5\%. The value of } k \text{ depends on the size of the document collection. We do not suggest that a large value of } k \text{ is used. This is because a large } k \text{ may result in unbalanced performance on different categories. In the future, a further study about the } k \text{ value sensitivity of the performance of classifiers will be conducted.}

We define our bottom-up hill climbing search algorithm as follows.

**Initial status:** current_concept_set \( \Phi_{ces} = \{\text{all leaf nodes in concept hierarchy}\}

\[
\text{heuristic} = f(\Phi_{ces})
\]

\[
\text{Temporal}_\text{concept_set} = \emptyset
\]

repeat

\[
\Phi_{ces} \leftarrow \Phi_{ces}
\]

\[
take \text{ a unmarked concept } c \in \Phi_{ces}
\]

- find parent concept node \( c_p \) of \( c \) in concept hierarchy
- use parent \( c_p \) to replace \( c \) in \( \Phi_{ces} \)
- remove all child concepts of \( c_p \) from \( \Phi_{ces} \)
- \( f = f(\Phi_{ces}) \)
- if \( f \geq f_{opt} \), then \( \Phi_{ces} \leftarrow \Phi_{ces} \), \( f_{opt} \leftarrow f \)
- else mark \( c \) in \( \Phi_{ces} \)

until no unmarked \( c \) in \( \Phi_{ces} \)

return: \( \Phi_{ces} \) as optimal concept set

### 3.4 Establish new Feature Model

Based on the optimum concept set obtained from the previous step, we can establish a new feature model for training documents. For new documents, we also map the terms to concepts using the same process. Then we add all the new concepts that did not occur in the training set to the concept hierarchy. We can now establish a concept description for test documents based on the feature model.

### 4. Experiment Setup and Results

In this section we experimentally evaluate the effect on the KNN classifier of using the domain-specific concept hierarchy to guide feature selection. In our experiments, we compare the performance of our feature selection method against the performance achieved by a common KNN classifier. Also, we study the effect of the training set size on the performance of both approaches.

The common KNN classification experiments used the RAINBOW system [18], which includes stop word removal, stemming and feature selection.

#### 4.1 Document Collection

We chose documents from 10 journals of the MEDLINE database [20] to form our training and test document sets as in table 1. Every document is labeled by the name of the journal that contains the document. The subjects of these documents are obviously independent of each other so they can be viewed as reasonable pre-existing categories. 150 documents were chosen randomly from each journal by people without specialized medical knowledge. 50 of 150 documents from each category were randomly chosen as test set and the remaining 1000 formed the training set except where otherwise specified. For one set of experiments comparatively studying the effect of the training set size, we formed specific-size subsets by randomly choosing the same number of documents from each category. The size of the test set was never varied in the experiments. We used title plus abstract as the text for our experiments.

#### 4.2 Accuracy Measure

To evaluate the trained classifier on test documents for each class, we defined an accuracy measure as follows. It is consistent with that used by RAINBOW system.
Accuracy = \frac{\text{correctly assigned documents}}{\text{total candidate documents}}

This accuracy can be used to measure the performance of the classifiers on each particular category. "correctly assigned documents" means all documents which are correctly assigned to the particular category. "total candidate documents" means all test documents which should be assigned to the particular category. The overall performance can be measured in the same way. Since every document in our data set has only one category label, the ‘recall’ measure is not considered in our experiments.

<table>
<thead>
<tr>
<th>Journal Name</th>
<th>Category Name</th>
<th>Covered years</th>
</tr>
</thead>
<tbody>
<tr>
<td>American Heart Journal</td>
<td>Heart</td>
<td>2000, 2001</td>
</tr>
<tr>
<td>Cancer Research</td>
<td>Cancer</td>
<td>1999, 2000</td>
</tr>
<tr>
<td>Bone</td>
<td>Bone</td>
<td>2000, 2001</td>
</tr>
<tr>
<td>Epilepsy research</td>
<td>Epilepsy</td>
<td>1999, 2000, 2001</td>
</tr>
<tr>
<td>Diabetes</td>
<td>Diabetes</td>
<td>1999, 2000</td>
</tr>
<tr>
<td>Clinical and experimental dermatology</td>
<td>Dermatology</td>
<td>1999, 2000</td>
</tr>
</tbody>
</table>

Table 1. Details of document collection

<table>
<thead>
<tr>
<th>Training Size</th>
<th>Distinct Terms</th>
<th>Distinct Concepts</th>
<th>Optimum Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>16,163</td>
<td>4,645</td>
<td>1,634</td>
</tr>
<tr>
<td>750</td>
<td>14,046</td>
<td>4,034</td>
<td>1,540</td>
</tr>
<tr>
<td>500</td>
<td>11,454</td>
<td>3,182</td>
<td>1,327</td>
</tr>
<tr>
<td>300</td>
<td>8,948</td>
<td>2,436</td>
<td>942</td>
</tr>
<tr>
<td>200</td>
<td>7,634</td>
<td>1,894</td>
<td>695</td>
</tr>
</tbody>
</table>

Table 2. Statistical information concerning the training set

<table>
<thead>
<tr>
<th>Category Name</th>
<th>Term-based</th>
<th>Original Concepts</th>
<th>Optimum Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addiction</td>
<td>100%</td>
<td>96%</td>
<td>94%</td>
</tr>
<tr>
<td>AIDS</td>
<td>92%</td>
<td>92%</td>
<td>90%</td>
</tr>
<tr>
<td>Heart</td>
<td>96%</td>
<td>92%</td>
<td>92%</td>
</tr>
<tr>
<td>Cancer</td>
<td>96%</td>
<td>90%</td>
<td>94%</td>
</tr>
<tr>
<td>Ophthalmology</td>
<td>72%</td>
<td>80%</td>
<td>84%</td>
</tr>
<tr>
<td>Burns</td>
<td>58%</td>
<td>66%</td>
<td>70%</td>
</tr>
<tr>
<td>Bone</td>
<td>66%</td>
<td>68%</td>
<td>74%</td>
</tr>
<tr>
<td>Epilepsy</td>
<td>74%</td>
<td>72%</td>
<td>78%</td>
</tr>
<tr>
<td>Diabetes</td>
<td>80%</td>
<td>82%</td>
<td>88%</td>
</tr>
<tr>
<td>Dermatology</td>
<td>30%</td>
<td>52%</td>
<td>58%</td>
</tr>
<tr>
<td>Overall</td>
<td>76.4%</td>
<td>79.0%</td>
<td>82.2%</td>
</tr>
</tbody>
</table>

Table 3. Comparison of accuracy in default training set

4.3. Document pre-processing

By pre-processing the training documents using the two methods separately, we derived statistical information about our training set as in table 2.

The number of distinct terms was obtained by using the mapping process and the number of optimum concepts was obtained by using the search algorithm we introduced above. For the heuristic function, \( \beta \) of 0.05 and \( k \) of 5 were used.

As we see in table 2, even using the original concept set, compared with the term set, causes a significant reduction in dimensionality of feature space.

4.4 Performance comparison between the two approaches

In this section, we used the default training set to compare the effect of different feature sets on performance of KNN classifier. Table 3 shows the
performance of the classifier on the 10 categories. A desirable classifier should have balanced performance for the pre-defined categories in the training set. Therefore, we computed the standard deviation (STD) for performance of the categories. It is possible that different values of k might be needed to achieve optimal performance for the different methods. For each method, therefore, we tried three values (5, 10, 15) of k, and the best results are reported in the result table.

A number of interesting observations can be made from the results in this table. First, compared with the term-based classifier, the overall performances achieved by the original concept model and the optimum concept model increased from 76.4% to 79% and 82.2% respectively, or a 3.4% and a 7.6% increase relatively respectively. Second, we see that our method smooths the performance of the classifier on different categories. Compared with the term-based classifier, the values of STD for two concept-based classifiers have a 31.4% and a 44.6% relative decrease respectively.

4.5 Effect of feature size on performance

In this section, we apply a feature selection method to documents in the pre-processing of a term-based KNN classifier. Through this experiment, we may study the effect of statistics-based feature selection methods on performance of term-based KNN classifier using the default training set. The RAINBOW system provides a feature selection function using the information gain method. A comparative study on feature selection methods [9], including document frequency thresholding (DF) [9], information gain (IG) [22], mutual information (MI) [9, 23], \( \chi^2 \) statistic (CHI) [9], and term strength (TS) [24, 25], shows that IG, DF and CHI have similar effects on performance of the classifiers and all are better than the other two. Therefore, the experimental results may, in a sense, provide information on how the statistics-based feature methods affect the performance of a KNN classifier on our dataset.

The influence of the information gain method is evaluated using the overall accuracy of the classifier and the STD of accuracies of individual categories. Figure 3 displays the two curves for the term-based KNN classifier on the default training set.

An observation emerges from the categorization results of figure 3. That is, although the performance of the classifier in terms of overall accuracy has not significantly declined until 90% of distinct terms are eliminated, the value of STD starts a clear increase once half the terms are removed. This means that we expect an imbalance of categorization accuracy when the information gain method is employed to reduce the dimensionality of the feature space for a KNN classifier on our dataset.

4.6 Effect of training set size on performance

A comparative experiment measuring the performance against the size of training set was conducted using training sets of different sizes listed in table 2. The optimum concept sets were discovered for training sets of different sizes. The feature selection algorithm was not used for this experiment because it does not improve the performance in terms of accuracy. Moreover, it is difficult to define a selection threshold for training sets of different sizes. The experimental results are shown in figure 4.

When the size of training set increased from 200 to 1000, the accuracy of concept-based classifier increased from 72.2% to 82.2%, or a 13.9% increase relatively, and the accuracy of term-based classifier increased from 66.8% to 76.4%, or a 14.4% increase relatively. In addition to this, another interesting observation can be made from this figure 4. We divide this process into two stages. In the first stage, when the size of training set increased from 200 to 500, the accuracy of the concept-based classifier increased from 72.2% to 79.8%, or a 7.6% increase relatively, and the accuracy of the term-based
classifier increased from 66.8% to 72.0%, or a 7.8% increase relatively. In other words, in this stage, the gradient of the accuracy of the concept-based classifier is 34.6% larger than that of the term-based classifier. In contrast, in the second stage, when the size of training set increased from 500 to 1000, the gradient of the accuracy of the term-based classifier is 103.3% larger than that of the concept-based classifier. It seems to indicate that the accuracy of the concept-based classifier converges faster than term-based classifier.

5. Discussion and Conclusion

The Experimental results show that seeking the optimum concept set in a concept hierarchical structure is a highly viable method. It enables us to reduce the length of document vectors effectively and reasonably. The experimental results also show that this approach, guided by the domain ontology, effectively improves the accuracy of a KNN classifier. Since we have so far conducted experiments in only a single domain, we cannot yet assert that this approach is suitable for all types of classification algorithms and document sets.

From the experimental results, two other impressive characteristics of ontology guided feature selection are

- more balanced performance on individual categories
- faster convergence of classification accuracy against the size of training set

When Information Gain is used to reduce the dimensionality of a feature space, the standard deviation of accuracies of individual categories has a faster rate of increase than the rate of decrease of overall accuracy. One experiment is not sufficient to make a judgment that all statistics-based feature selection methods will lead to unbalanced performance on individual categories in all datasets. However, all statistics-based feature selection methods have a common feature. They involve searching for an optimum subset of features based on term-goodness criteria (e.g. information gain). Thus their effectiveness depends on the quality of training set. In a sense, feature selection problem can be viewed as a special case of the feature weighting problem. No matter which feature selection method is applied, accurately assigning a score to a distinct term depends on whether enough information is provided by the training set. In most situations, the key terms for the individual categories of different natures are assigned scores distributed in a large range. Due to noise or insufficiency of training documents, some non-informative terms may be assigned high weights. Therefore, some terms important to identify the content of some categories are removed earlier than the key terms of other categories and even some non-informative terms. Rather than simply removing terms, the ontology guided feature selection uses the parent concept to replace child concepts that have the same nature, and is thus less affected by noise.

KNN is an instance-based classifier. The performance of instance-based classifiers is more dependent on sufficiency of training set than that of other machine learning classification algorithms. KNN treats documents as individual points in a vector space. Each point is defined by a set of non-zero features of feature space. A smaller training set implies that more terms or term combinations important for content identification may be missing from the training documents. This will negatively affect the performance of a classifier. The domain ontology guided approach can somewhat reduce the negative influence of this problem. We discover the optimum concept set in the concept hierarchy by replacing child concepts with parent concepts when this does not adversely affect performance. Therefore an important term, which resides low in the concept hierarchy, may be mapped to a concept in the optimum concept set, even if this term is not included in the training set.

Acknowledgments

We would like to thank the U.S. National Library of Medicine for having provided the UMLS knowledge resources.

References:


