Journal of Biomedical Informatics 43 (2010) 31–40

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

GOClonto: An ontological clustering approach for conceptualizing PubMed abstracts

Hai-Tao Zheng [a,b], Charles Borchert [a], Hong-Gee Kim [a,*]

[a] Biomedical Knowledge Engineering Laboratory, BK21 College of Dentistry, Seoul National University, 28 Yeongeon-dong, Jongro-gu, Seoul 110-749, Republic of Korea
[b] Graduate School at Shenzhen, Tsinghua University, Shenzhen, PR China

ARTICLE INFO

Article history:
Received 31 October 2008
Available online 25 July 2009

Keywords:
GOClonto
PubMed abstract
Ontological clustering
Gene ontology
Conceptualization
Ontology generation
Suffix tree clustering
Lingo
Fuzzy Ants clustering
Tolerance rough set

ABSTRACT

Concurrent with progress in biomedical sciences, an overwhelming of textual knowledge is accumulating in the biomedical literature. PubMed is the most comprehensive database collecting and managing biomedical literature. To help researchers easily understand collections of PubMed abstracts, numerous clustering methods have been proposed to group similar abstracts based on their shared features. However, most of these methods do not explore the semantic relationships among groupings of documents, which could help better illuminate the groupings of PubMed abstracts. To address this issue, we proposed an ontological clustering method called GOClonto for conceptualizing PubMed abstracts. GOClonto uses latent semantic analysis (LSA) and gene ontology (GO) to identify key gene-related concepts and their relationships as well as allocate PubMed abstracts based on these key gene-related concepts. Based on two PubMed abstract collections, the experimental results show that GOClonto is able to identify key gene-related concepts and outperforms the STC (suffix tree clustering) algorithm, the Lingo algorithm, the Fuzzy Ants algorithm, and the clustering based TRS (tolerance rough set) algorithm. Moreover, the two ontologies generated by GOClonto show significant informative conceptual structures.

1. Introduction

As biomedical science progresses, bio-engineering and functional genomics has lead to a vast amount of research. The broadening of new research fields causes an exponential growth in the amount of biomedical literature. PubMed [1] is the most comprehensive database collecting and organizing biomedical literature. Since gene ontology (GO) [2] provides a controlled vocabulary to describe gene and gene product attributes in any organism, there are numerous methods that attempt to exploit biomedical literature through PubMed using text mining or machine learning techniques based on GO. Raychaudhuri et al. [3] proposed maximum entropy to associate a set of GO codes to PubMed abstracts and thus to the genes associated with the abstracts. Theodosiou et al. [4] used linear discriminant analysis (LDA) to classify PubMed abstracts for functionally annotating genes. Izumitani et al. [5] proposed support vector machine (SVM) and maximum entropy method (MEM) for assigning upper level gene ontology terms to genes using relevant documents. Chen et al. [6] also proposed an automated linking scheme for PubMed abstract with GO-terms using SVM. Vanteru et al. [7] introduced latent semantic analysis (LSA) to link the PubMed abstracts to the GO, called SEGOPubmed, for ontology-based browsing. GOPUBMED [8,9] was proposed as a web server which allows users to explore PubMed search results with the gene ontology. GO-KDS [10] uses a machine learning technique, called the weighted confidence learner (WCL), to find the closely matching genes or proteins in GO from PubMed abstracts.

Recently, many clustering methods have been proposed that can help mitigate the training paradigm of supervised learning, especially in cases where the partitioning of the document space is not known a priori. In addition, computational methods for clustering PubMed documents are required to help domain experts such as biologists or medical scientists effectively retrieve PubMed documents relevant to their interests. To this end, a number of clustering algorithms that extract meaningful labels from documents have been developed to help users better understand the structure of document collections. Zamir et al. [11] proposed a phrase-based document clustering approach based on suffix tree clustering (STC). Schockaert [12] developed a clustering method using Fuzzy Ants, which uses ant colony optimization principles to find good partitions of the data. Lang [13] presented an algorithm for web search results clustering based on tolerance rough set (TRS), which is able to deal with vagueness and fuzziness and is used to model relations between terms and documents. Osinski et al. [14] proposed a concept-driven algorithm for clustering search results, the Lingo algorithm, which uses the latent semantic indexing (LSI) technique to separate search results into meaningful groups. Zheng et al. [15] exploited noun phrases and semantic relationships to cluster text documents. In the biomedical domain, TextQuest [16] was presented to cluster PubMed abstracts using

* Corresponding author.
E-mail address: hgkim@snu.ac.kr (H.-G. Kim).

1532-0464/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2009.07.006

the *k*-means algorithm. MeSHer [17] uses a simple statistical approach to identify biological concepts in the form of medical subject headings (MeSH terms) obtained from the PubMed database that are significantly overrepresented within the identified gene set relative to those associated with the overall collection of genes on the underlying DNA microarray platform. Yamamoto et al. [18] developed a system called McSyBi to hierarchically and non-hierarchically cluster PubMed abstracts. Homayouni et al. [19] explored LSI to automatically identify gene relationships from titles and abstracts in PubMed. Lin et al. [20] developed an approach that retrieved and organized PubMed abstracts into different topical groups and prioritized important citations in each group. Theodosiou et al. [4] proposed a graph-based PubMed abstract clustering methodology called PuRed-MCL, which is based on the Markov clustering algorithm (MCL).

However, most of the existing clustering methods are focused on grouping documents only; they do not explore the semantic relationships of document groupings. Semantic relationships are defined as any relationship between two or more concepts based on the meaning of the concepts [21]. Exploiting the semantic relationships of document groupings not only helps users visualize and comprehend the underlying structures of document collections, but also enables computers to perform the inference process for retrieving documents based on user queries more accurately. Although hierarchical clustering methods are presented in existing methods [18,19], they do not maintain any semantics of relationships between groupings. To address this issue, we propose an ontological clustering method called GOClonto for conceptualizing PubMed abstracts. GOClonto utilizes latent semantic analysis (LSA) and gene ontology (GO) to identify key gene-related concepts and their relationships as well as allocate PubMed abstracts based on these key gene-related concepts. In this study, conceptualization of PubMed abstracts means representing PubMed abstracts with a set of key gene-related concepts and their relationships, which can help users understand PubMed abstract collections through intuitive, structured, semantic connections between the gene-related concepts. Since gene-related concepts extracted by GOClonto are contained in GO, we call them GO-terms. Ontological clustering is defined as a method that not only clusters documents, but also explores the ontologically based semantic relationships between the clusters. Key GO-terms are defined as the most important gene-related concepts to which a PubMed abstract collection is related. GOClonto has a number of advantages:

1. It identifies the key GO-terms of a PubMed abstract collection, a simplified and relevant list of terms for the collection.
2. It generates a corpus-related ontology, closely related to the collection, but significantly smaller than GO and more manageable. The result ontology is a simplified and clear conceptual structure of the key GO-terms and their relations, laid out on in OWL format [22], which enables flexible functionality, such as DL reasoning (description logic reasoning).
3. It allows browsing of PubMed abstract collections by key GO-terms—LSA utilization creates overlapping groups of allocated documents based on the key GO-terms, so all documents explicitly and implicitly related to a key GO-term are allocated appropriately, and are thus able to be browsed when relevant.

2. Methods

The general idea of GOClonto is to automatically generate a corpus-related gene ontology, which represents the conceptual structure of a PubMed abstract collection. Fig. 1 shows the overview of the GOClonto method. Specifically, GOClonto involves the following main steps:

1. A PubMed abstract collection is preprocessed into term frequency files, in which each abstract is represented as a list of its term frequencies.
2. Based on GO, GO-terms in the collection are identified and stored.
3. LSA techniques are used to perform key GO-term induction and related document allocation.
4. A corpus-related gene ontology is generated to maintain the semantic relationships between key GO-terms. Then, PubMed abstracts are linked to the corpus-related ontology through these key GO-terms.

2.1. Preprocessing and GO-term identification

At the preprocessing step, we first conduct tokenization to split a document into sentences. Based on a stopword list built by Gerard Salton and Chris Buckley [23], we remove the words that occur frequently but have no meaning. Second, we perform POS tagging using CRFtagger [24], a Java-based conditional random fields POS Tagger for English. Third, we utilize the stemming function provided by WordNet [25] to perform word stemming. Finally, all the nouns contained in each PubMed abstract are counted and used to compose the term frequency file of each PubMed abstract.

To identify GO-terms, we need to recognize noun phrases in addition to the nouns. CRFChunker [26], a Java-based conditional random fields phrase chunker, is employed to identify noun phrases. With the identified nouns and noun phrases, GOClonto determines whether or not the nouns or noun phrases are GO-terms by referencing GO, i.e., GOClonto checks whether or not the nouns or noun phrases are contained in GO. If the nouns or noun phrases are contained in GO, we recognize them as GO-terms and store them.

To illustrate GOClonto, we use a simple example collection of $d = 8$ biomedical documents (Fig. 2(a)), in which $t = 6$ nouns (Fig. 2(b)) appear more than once and thus are treated as frequent. In addition, we can see that $g = 6$ GO-terms are extracted by GOClonto, which consist of not only single-word GO-terms, but also multi-word GO-terms (Fig. 2(c)).

2.2. Key GO-term induction and related document allocation

The intuition of key GO-term induction is that key GO-terms should have more closely related documents than that of other GO-terms in the collection. Before applying LSA to perform key GO-term induction, we need to construct the term–document matrix. The *tfidf* (term frequency-inverted document frequency) is applied to calculate the weights of terms. In the vector space model, a document d is represented as a feature vector $\vec{d} = (tf_{t_1,d}, \ldots, tf_{t_i,d})$, where $tf_{t,d}$ returns the absolute frequency of term $t \in \mathscr{T}$ in document $d \in \mathscr{D}$, where $\mathscr{D}$ is the document collection and $\mathscr{T} = \{t_1, t_2, \ldots, t_i\}$ is the set of unique terms occurring in $\mathscr{D}$. To weigh the frequency of a term in a document with a factor that discounts its importance when it appears in many documents, the *idf* (inverted document frequency) of term t in document d is proposed by [27] as follows:

$$idf_t = log(n/df_t) \quad (1)$$

where n is the total number of documents in the collection and df_t is the document frequency of term t that counts how many documents in which term t appears. Consequently, the *tfidf* measure is calculated as the weight $w_{t,j}$ of term t in document j:

$$w_{t,j} = tf_{t,j} \times idf_t \quad (2)$$

With the weight $w_{t,j}$ of term t, we can construct the term–document matrix. For the example we used (Fig. 2), after calculating the term

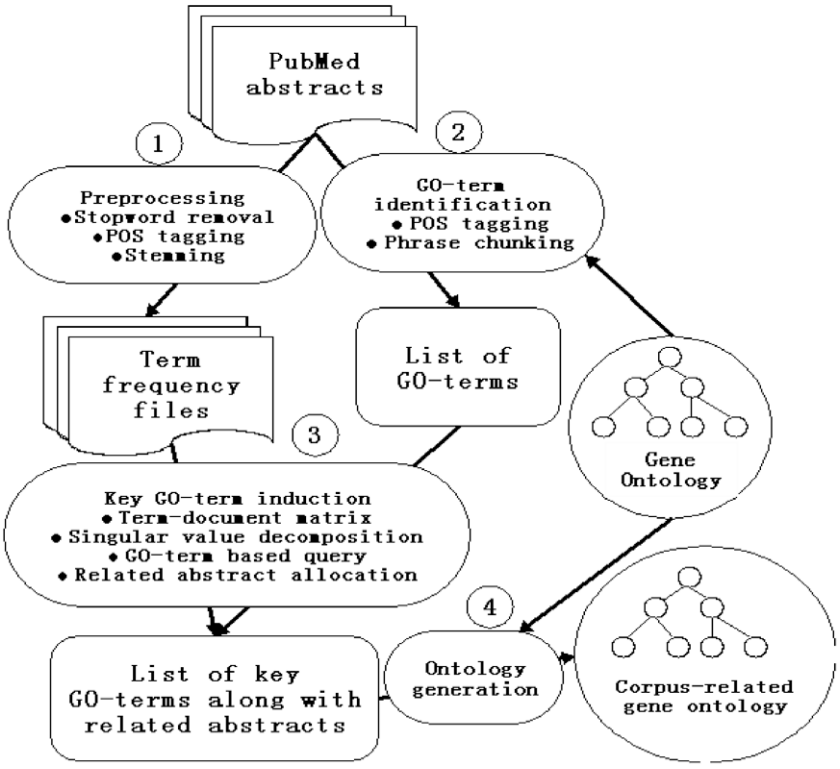

Fig. 1. An overview of the GOClonto method.

weights, each document is represented as a feature vector, which is used to compose the term–document matrix **A**. Then, we normalize each column vector's length of matrix **A** as shown in Fig. 3(a). In this matrix, each column vector represents each document, and each row vector denotes each term extracted to represent the documents' features. In our example, the first row represents the term T1 'cell', the second row represents the term T2 'membrane' and so on through the terms listed in Fig. 2(b). Similarly, the columns denote the documents listed in Fig. 2(a). The first column represents document D1, the second column represents document D2, and so on.

To conduct the key GO-term induction, we apply LSA to process the term–document matrix by performing the singular value decomposition (SVD) of matrix **A**, which breaks it into three matrices (**U**, **S**, and **V**) such that $\mathbf{A} = \mathbf{USV}^T$. In this way, SVD translates the term and document vectors into a concept space. The first r columns of **U** and **V** (where r is **A**'s rank) form an orthogonal basis for the term–document matrix's term space and document space, respectively. One advantage of LSA is that it finds a low-rank approximation to the term–document matrix and removes noise [28]. When we select the k largest singular values from **S** (Fig. 3(b)), and their corresponding singular vectors from **U** (Fig. 3(c)) and **V** (Fig. 3(d)), we get the rank k approximation $\widehat{\mathbf{A}}$ to **A** with the smallest error in terms of Frobenius norm, that is, $\widehat{\mathbf{A}} = \mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T$ is the best square approximation of **A** by a matrix of rank k in the sense defined in the equation $\triangle = \|\mathbf{A} - \widehat{\mathbf{A}}\|_F$.

In practice, we determine the most significant k singular values by selecting the Frobenius norms of the matrix **A** and its k-rank approximation $\widehat{\mathbf{A}}$. The Frobenius norm measures the difference between two matrices. Let threshold p be a percentage-expressed value that determines to what extent the k-rank approximation should retain the original information in matrix **A**. We

(a)
D1: All of the contents of a cell excluding the plasma membrane and nucleus.
D2: The lipid bilayer surrounding an organelle.
D3: A septum which spans a cell and does not allow exchange of organelles or cytoplasm between compartments.
D4: Caveolaes may be pinched off to form free vesicles within the cytoplasm.
D5: A cell junction at which the cytoplasmic face of the plasma membrane is attached to actin filaments.
D6: The process by which cells digest parts of their own cytoplasm.
D7: A cellular organelle, found close to the nucleus in many eukaryotic cells.
D8: The change in shape of the spermatid nucleus from a spherical structure to an elongated organelle.

(b)
T1: Cell
T2: Membrane
T3: Nucleus
T4: Plasma
T5: Organelle
T6: Cytoplasm

(c)
GO-T1: Cell
GO-T2: Plasma membrane
GO-T3: Nucleus
GO-T4: Organelle
GO-T5: Cytoplasm
GO-T6: Cell junction

Fig. 2. A simple biomedical document collection example.

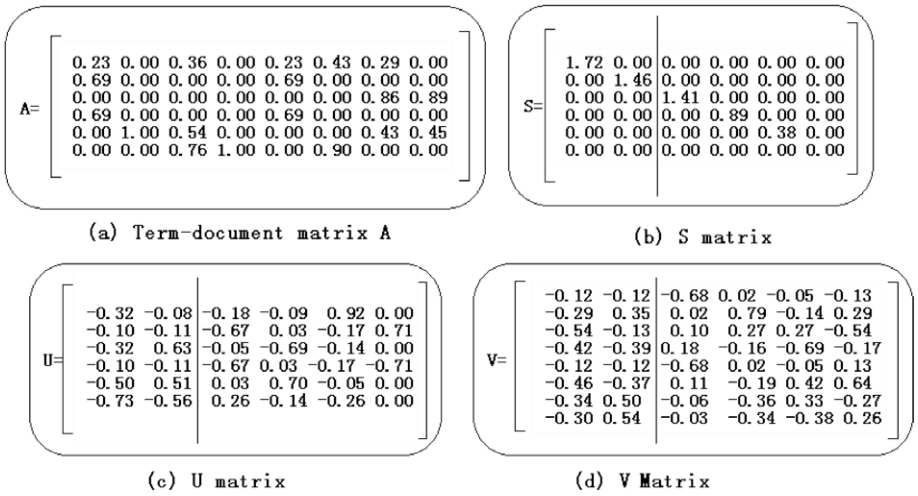

Fig. 3. Matrices used for the key GO-term induction.

consequently define k as the minimum value that satisfies the following condition:

$$\|\widehat{\mathbf{A}}\|_F / \|\mathbf{A}\|_F \geqslant p \tag{3}$$

where the symbol $\|\mathbf{X}\|_F$ denotes the Frobenius norm of matrix $\mathbf{X}$. The larger the value of p the larger k will be used. For our example, threshold $p = 0.5$ is used. Consequently, value of k is determined as 2.

To identify key GO-terms based on matrix $\widehat{\mathbf{A}}$, GO-terms should be treated as queries and compared to documents. For this purpose, first, we treat identified GO-terms as pseudo-documents, i.e., a pseudo-document is composed of a GO-term. Then, we construct the query vectors of these pseudo-documents with *tfidf* weight, i.e., a query vector $\vec{q_0} = (w_{t_1,q}, \ldots, w_{t_i,q})$. Since the documents are represented as row vectors in the document matrix $\mathbf{V}_k$, a GO-term query must be represented as a vector in k-dimensional space and compared to the document vectors. Suppose $\vec{q_0}$ is a query vector of a GO-term, a new query vector $\vec{q}$ is computed as:

$$\vec{q} = \vec{q_0}^T \mathbf{U}_k \mathbf{S}_k^{-1} \tag{4}$$

Next, we use cosine similarity to compute the similarity scores between queries and documents as:

$$sim(\vec{q}, \vec{d}_{k,j}) = \frac{\vec{q} \cdot \vec{d}_{k,j}}{|\vec{q}||\vec{d}_{k,j}|} \tag{5}$$

where $\vec{q}$ is the new query vector of a GO-term and $\vec{d}_{k,j}$ is the jth row vector contained in the document matrix $\mathbf{V}_k$. To illustrate the similarity calculation process, we first calculate the query vector of the GO-T2 'plasma membrane' in the example and obtain $\vec{q_0} = (0.00, 1.00, 0.00, 1.00, 0.00, 0.00)^T$. The new query vector in k-dimension ($k = 2$ in the example) is computed as:

$$\vec{q} = (0.00, 1.00, 0.00, 1.00, 0.00, 0.00) \cdot \begin{pmatrix} -0.32 & -0.08 \\ -0.10 & -0.11 \\ -0.32 & 0.63 \\ -0.10 & -0.11 \\ -0.50 & 0.51 \\ -0.73 & -0.56 \end{pmatrix} \cdot \begin{pmatrix} 1.72 & 0.00 \\ 0.00 & 1.46 \end{pmatrix}^{-1} = (-0.11 - 0.16)$$

Next, the similarity between GO-T2 'plasma membrane' and document D1 is calculated as:

$$\begin{aligned} sim(\vec{q}, \vec{d}_{k,1}) &= \cos((-0.11, -0.16), (-0.12, -0.12)) \\ &= \frac{(-0.11) * (-0.12) + (-0.16) * (-0.12)}{\sqrt{(-0.11)^2 + (-0.16)^2} \cdot \sqrt{(-0.12)^2 + (-0.12)^2}} \\ &= 0.9832 \end{aligned}$$

According to a GO-term query vector $\vec{q}$, we sum up all the $sim(\vec{q}, \vec{d}_{k,j}), j = 1, \ldots, n$, and get the total similarity score s of the corresponding GO-term. Then, all the GO-terms are sorted based on their similarity score s in decreasing order. If we select key GO-terms based on the similarity score s only, it may result in a local optimization problem, i.e., only a group of similar GO-terms with high score s would be used to represent the collection. To overcome this problem, for each two GO-terms x and y, we calculate the cosine similarity of their corresponding query vectors $sim(\vec{q}_x, \vec{q}_y)$. By setting a term similarity threshold f, we group all the GO-terms in which each pair of them have similarity $sim(\vec{q}_x, \vec{q}_y) > f$. In this study, the value of f is set as 0.85 because two vectors can be considered to have high similarity if their cosine similarity is higher than 0.85. The GO-term with the highest score s in each grouping are selected. Among them, the first m GO-terms are determined as key GO-terms. In addition, if a GO-term frequently occurs in the collections and is found to be a subclass of a key GO-term, it is included as a key GO-term. This process is to find the GO-terms which give more specific meaning than the identified GO-terms. As a result, $m = 3$ key GO-terms are selected, which are GO-T1 'cell' with $s = 5.48$, GO-T5 'cytoplasm' with $s = 4.43$, and GO-T2 'plasma membrane' with $s = 3.62$.

During the query process in LSA, for a GO-term, we obtain a list of documents satisfying $sim(\vec{q}, \vec{d_{k,j}}) > e$, where e is the document similarity threshold and $\vec{q}$ is the query vector of the GO-term. The documents satisfying $sim(\vec{q}, \vec{d_{k,j}}) > e$ can be considered closely related to the corresponding GO-term. Consequently, we allocate the related documents for each GO-term. In our example, e is set as 0.9 and the allocation results are shown in Fig. 4. This assignment method allows naturally creates overlapping groups and handles cross-topic documents well. Moreover, even if a document which is closely related to a GO-term does not contain that GO-term, it can be allocated to the GO-term related group because we utilize LSA to analyze the similarities between GO-terms and documents. For instance, document D4, which is semantically related to GO-term 'cell' implicitly, is allocated to GO-term 'cell' related group in our example. For a given GO-term, the semantically related documents are important to biologists even when some

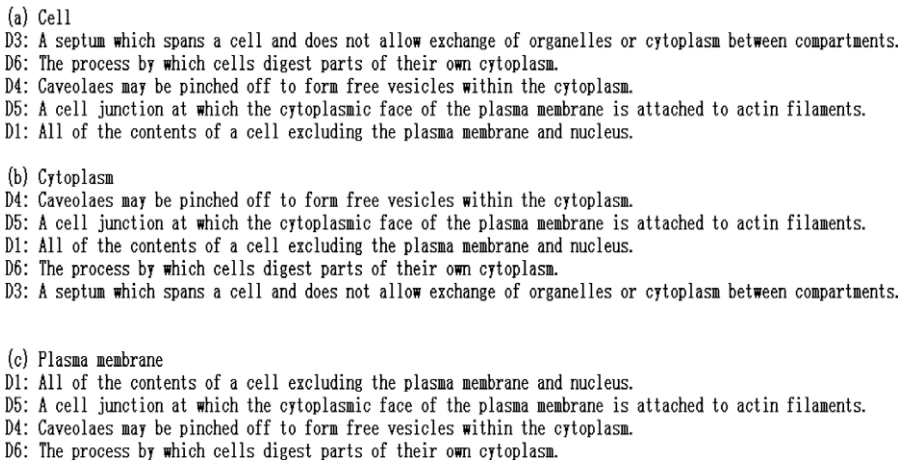

Fig. 4. Related document allocation results for each key GO-term.

documents do not contain the GO-term explicitly. For example, some gene products for a GO-term do not contain the GO-term while these gene products are described in the related scientific documents. These documents can help biologists understand the specific gene products for a GO-term in specific experimental environments. In addition, these documents can also be used as evidence in gene ontology annotation work [29], which is proposed to build the connection between a type of gene product and the types designated by terms in an ontology such as the GO.

2.3. Corpus-related ontology generation

To generate an ontology based on a set of key GO-terms, we develop an algorithm called Corpus-related gene ontology generation algorithm (Algorithm 1). This algorithm uses a set of key GO-terms and their subclass GO-terms, which are identified among the frequent GO-terms in a document collection, as input. The basic idea of the algorithm is to identify the common superclass GO-term of all the input GO-terms and store the subtree, whose root node is the superclass GO-term, as a corpus-related gene ontology. Note that Delfs et al. [9] also presented an algorithm to create ontologies by starting at GO-terms and iteratively look up parent GO-terms until the root of GO is reached. The ontologies generated by our algorithm are smaller and more manageable than Delfs's, because our algorithm only identifies the common superclass GO-term of all the input GO-terms and utilized it to generate the corpus-related gene ontology. However, Delfs's algorithm looks up all the parent GO-terms of input GO-terms until the root of GO is reached, so that large amount of classes will be created in the generated ontologies when most of the input GO-terms are leaf nodes in the GO.

In the Corpus-related gene ontology generation algorithm (Algorithm 1), a tree structure is used to store GO-terms and their subclass GO-terms, each tree node representing a GO-term, and its subclass GO-terms stored as subnodes of this tree node. α is a list of tree nodes storing GO-terms, initially containing the original input. For each iteration, a tree node t_j whose GO-term is not the root in GO is selected from α. The direct superclass pr of t_j is obtained and checked for presence in α by recursively examining all the tree nodes and their subnodes in α. If a GO-term in GO has multiple parent GO-terms, the corresponding tree node pr will be created for each parent GO-term. If pr is not contained in α, pr is added to α. Next, the tree node t_j is added as pr's subnode. t_j is removed from α because t_j has been added as a subnode of pr. Finally, when the common superclass GO-term of all the input GO-terms is found, the tree node having this common superclass GO-term as its root, the last item in α, represents the generated ontology. GOClonto recursively stores the whole tree into an OWL file [22]. For instance, since GO-T1 'cell', GO-T5 'cytoplasm and GO-T2 'plasma membrane' are identified as key GO-terms in the example, they are used as input to generate a corpus-related ontology. When the common superclass GO-term 'cellular component' is recognized, our algorithm generate the subtree of GO, whose root node is GO-term 'cellular component', as the corpus-related gene ontology (Fig. 5).

Fig. 5 shows the generated ontology of our example in the user interface of the tool 'GOClonto', which helps users conceptualize a PubMed abstract collection by automatically generating a corpus-related gene ontology. The documents, having been allocated to their related key GO-terms, are then linked to the ontology through these GO-terms. When users select GO-terms in the ontology, their corresponding documents automatically display in the right panel. Note that all of the documents allocated to a GO-term's subclass GO-terms are also allocated to that GO-term. A conceptual structure of the biomedical document collection in the example is clearly represented in the generated ontology. The documents allocated to GO-T2 'plasma membrane' are also related to the documents allocated to GO-T5 'cytoplasm', because the two GO-terms have the same superclass GO-term 'cell part'. GO-term 'cell part' incorporates all documents allocated to GO-T2 'plasma membrane' and GO-T5 'cytoplasm'. In addition, since all the GO-terms have the same superclass 'cellular component', users can see that the whole document collection is related to this more general GO-term. Therefore, with this ontology, users not only see the potential GO-terms related to the document collection, but can also more clearly see the semantic relationships between groupings of the documents.

Algorithm 1. Corpus-related gene ontology generation algorithm

Input: $\eta \leftarrow$ a set of key GO-terms and their subclass GO-terms
$\alpha \leftarrow$ Empty list {α is a list of tree nodes storing GO-terms used to construct the ontology}
for each GO-term g_i in η **do**
  Create tree node t_i that represents g_i
  Add t_i to α
end for
while α has more than one tree nodes **do**
  Get a tree node t_j from α whose GO-term is not the root in GO
  Get the direct superclass GO-term p of t_j's GO-term from GO
  Create tree node pr that represents p
  if pr is not found in α **then**
    Add pr to α
  end if
  Set t_j as subnode of pr
  Remove t_j from α
end while
Output the last tree node in α as a corpus-related gene ontology $\mathcal{O}$ in OWL
format

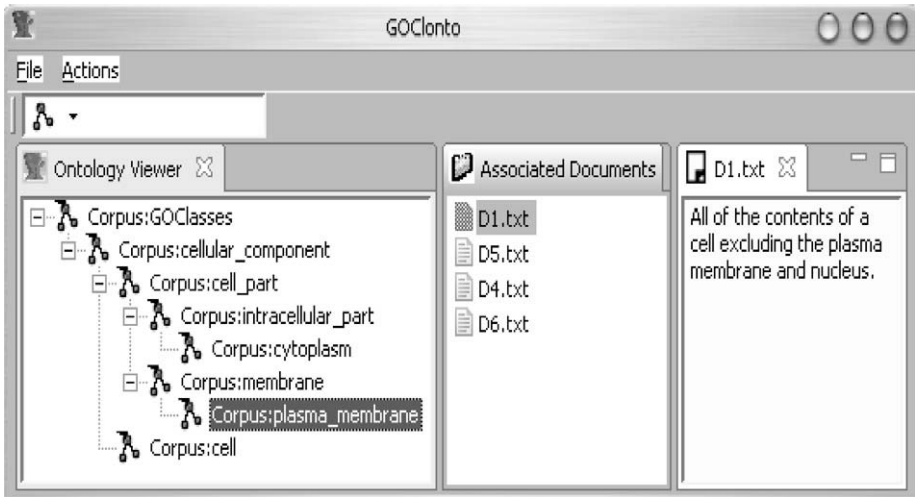

Fig. 5. Generated ontology of the document collection example in GOClonto.

3. Results

3.1. Experiment setup

To examine the effectiveness of GOClonto, we conducted control experiments based on two different datasets. First, to evaluate the results of key GO-term identification, we combined documents that belong to pre-defined categories and examined whether or not GOClonto can identify the category topics as key GO-terms. In addition, we compared the results with that of the suffix tree clustering (STC) algorithm [11], the Lingo algorithm [14], the Fuzzy Ants clustering algorithm [12], and clustering based on tolerance rough set (TRS) [13]. Second, to evaluate the effectiveness of document allocation of each key GO-term, we performed a clustering evaluation by comparing GOClonto with the above four clustering algorithms in terms of *F*-measure. Finally, to evaluate informativeness of the generated ontology, we compared the ontology generated by GOClonto with the hierarchical tree generated by the Fuzzy Ants clustering algorithm [12]. The experiments were performed on J2SE 5.0, Windows XP, Pentium 4, 3.0 GHz with 2GB RAM.

We collected document sets related to various GO-terms from PubMed. We used the 'MajorTopic' tag along with the GO-terms as queries to PubMed. Since the retrieved documents are tagged manually with GO-terms as a result of common sense agreement of many users, we use them as the answer set for experiments. Since PubMed abstracts are annotated by MeSH headings [30], the GO-terms we selected are contained in MeSH headings. We checked the definitions of the selected GO-terms in MeSH to guarantee these concepts having similar semantics with their definitions in GO. The GO-terms used for the queries were also used as category names. Since the documents are retrieved based on these GO-terms, we treat these significant GO-terms as target key GO-terms to evaluate whether or not our method can identify key GO-terms. For each category, documents were assembled from the titles and abstracts retrieved from PubMed. The first dataset is composed of five categories: organelle, chromosome, cytoskeleton, kinetochore, mitochondria, with totally 324 PubMed abstracts (Table 1). The second dataset is composed of four categories: vacuole, phagolysosome, acrosome, and lysosome with totally 334 PubMed abstracts (Table 2).

To evaluate the quality of the clustering results, we adopted a quality measure, *F*-measure, which is widely used in the text mining literature for the purpose of document clustering [31]. Note that since GOClonto creates overlapping groups during clustering process, the conventional clustering measurements, entropy [31] and purity [32], are not suitable to be used in this study. *F*-measure combines the precision and recall ideas found in the information retrieval literature. Each cluster is treated as if it were the result of a query and each class is treated as if it were the desired set of documents for a query. The precision and recall of a cluster j with respect to a class i are defined as:

$$\mathscr{P} = Precision(i,j) = \frac{n_{ij}}{n_j} \tag{6}$$

$$\mathscr{R} = Recall(i,j) = \frac{n_{ij}}{n_i} \tag{7}$$

where n_{ij} is the number of members of class i in cluster j, n_j is the number of members of cluster j and n_i is the number of members of class i. The *F*-measure of cluster j and class i is then given by $\mathscr{F}(i,j) = 2\mathscr{P}\mathscr{R}/(\mathscr{P}+\mathscr{R})$. The overall *F*-measure is computed by taking all the values for the highest *F*-measure of each class, which is normalized by the class size as the following:

$$\mathscr{F} = \sum_i \frac{n_i}{n} \max\{\mathscr{F}(i,j)\} \tag{8}$$

where $\max\{\mathscr{F}(i,j)\}$ is the highest *F*-measure for a given class among all the values of *F*-measure of clusters to that class, n is the number of documents.

3.2. Experiment results

According to different values of thresholds p, f, and e, different clusters are generated and different numbers of documents are allocated based on their related GO-terms. The research in [33]

Table 1
First PubMed abstract dataset.

Category	Number of documents	Description
Organelle	24	Abstracts of biomedical literature related to organelle
Chromosome	75	Abstracts of biomedical literature related to chromosome
Cytoskeleton	75	Abstracts of biomedical literature related to cytoskeleton
Kinetochore	75	Abstracts of biomedical literature related to kinetochore
Mitochondria	75	Abstracts of biomedical literature related to mitochondria

Table 2
Second PubMed abstract dataset.

Category name	Number of documents	Description
Vacuole	85	Abstracts of biomedical literature related to vacuole
Phagolysosome	79	Abstracts of biomedical literature related to phagolysosome
Acrosome	85	Abstracts of biomedical literature related to acrosome
Lysosome	85	Abstracts of biomedical literature related to lysosome

shows that when threshold p is between 0.12 and 0.25, LSA is able to remove noise and achieve relatively good performance. Consequently, we selected the values of p between 0.12 and 0.25 in our experiments. Since two vectors have high similarity if their cosine similarity is higher than 0.85, we consider the values of threshold f between 0.85 and 0.95. If the threshold e is too large, the recall of clustering will decrease. If the threshold e is too small, the precision of clustering will decrease. Based on the two datasets, we conduct the experiments to choose a suitable value of threshold e, which are shown in Figs. 6 and 7. We found that F-measure values achieve high values when e is between 0.3 and 0.5. Therefore, we applied such e to our experiments. Note that although a complicated algorithm can be developed to estimate the parameter values in order to obtain the optical values of F-measure, this algorithm will increase the complexity and decrease the efficiency of our method greatly. Therefore, since our method with these predefined parameter values can obtain acceptable values of F-measure, we used these parameter values to achieve a trade-off between precision and complexity. In addition, since all the parameters are hidden from users, users do not need to be familiar with the task of parameter setting. The experimental results were compared with other clustering algorithms. The desired clusters of other algorithms were set from 3 to 15. We tested each algorithm with various parameters and chose the best clustering results.

Based on the two PubMed abstract datasets, the key GO-terms extracted by GOClonto and the other four clustering algorithms are listed in Table 3. With respect to the first dataset, GO-Terms 'microtubule' and 'microtubule cytoskeleton' were identified by GOClonto because they are closely related to GO-term 'cytoskeleton', i.e., microtubule is part of a cytoskeleton and microtubule cytoskeleton is a kind of cytoskeleton. Similarly, the identified GO-terms 'kinetochore microtubule' and 'mitochondrial translation' are closely related to GO-terms 'kinetochore' and 'mitochondria', respectively. GO-term 'organelle' was not induced because all the abstracts in the first dataset are related to this term so GOClonto does not distinguish it. For the other four clustering algorithms, the STC algorithm did not identify any key GO-terms; the Lingo algorithm recognized GO-terms 'unattached kinetochore' and 'sex chromosome' only; the Fuzzy Ants algorithm

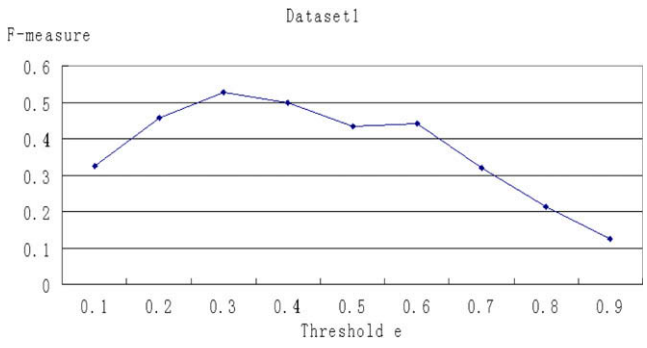

Fig. 6. Different F-measure values based on threshold e for dataset1.

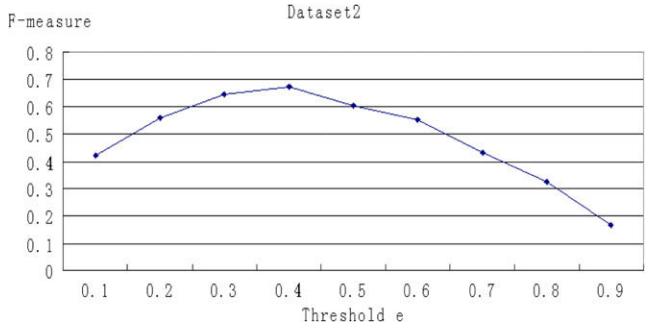

Fig. 7. Different F-measure values based on threshold e for dataset2.

distinguished GO-term 'cytoskeleton' only; and the Clustering based on TRS algorithm identified GO-terms 'kinetochore' and 'cytoskeleton' only.

With respect to the second dataset, all the key GO-terms were identified by GOClonto, i.e., GO-terms 'vacuole', 'lysosome', 'phagolysosome', and 'acrosome' were all recognized. In addition, not only these key GO-terms, but also the other GO-terms closely related to these key GO-terms, such as GO-term 'lytic vacuole', were identified, giving a more specific meaning than the key GO-term 'vacuole'. This shows how the GOClonto includes not simply the key GO-terms used to distinguish document groups, but also uses other important GO-terms that may be helpful to navigate the whole document collection. According to the other four clustering algorithms, the STC algorithm recognized GO-term 'lysosome' only; the Lingo algorithm identified GO-terms 'sperm acrosome', 'mycobacterial phagosomes', and 'parasitophorous vacuoles' only; the Fuzzy Ants algorithm did not distinguish any key GO-terms; and the Clustering based on TRS determined GO-term 'autophagic vacuoles' only. Therefore, we can see above that GOClonto is able to recognize key GO-terms from the PubMed abstract collections and outperforms the other four clustering algorithms in terms of key GO-term identification.

Table 4 shows the F-measure values of all the clustering methods. With respect to the first dataset, GOClonto has the highest F-measure value 0.5294. The F-measure of GOClonto is 0.141 higher than STC, 0.1468 higher than Lingo, 0.3091 higher than Fuzzy Ants, and 0.3696 higher than clustering based on TRS. With respect to the second dataset, GOClonto has the highest F-measure value 0.6746. The F-measure of GOClonto is 0.1047 higher than STC, 0.0136 higher than Lingo, 0.2744 higher than Fuzzy Ants, and 0.5872 higher than clustering based TRS. Therefore, we can see that GOClonto outperforms the other four clustering algorithms in terms of F-measure, which means GOClonto can allocate PubMed abstracts to their related GO-terms with a relatively high precision.

Based on the first dataset, a corpus-related gene ontology generated by the GOClonto method is shown in Fig. 8 and a hierarchical tree created by the Fuzzy Ants clustering algorithm is shown in Fig. 9. Based on the second dataset, a corpus-related gene ontology generated by the GOClonto method and a hierarchical tree created by the Fuzzy Ants clustering algorithm are shown in Figs. 10 and 11, respectively. We found that the ontology is much more informative than the hierarchical tree. GO is structured as directed acyclic graphs and many GO-terms are inherited from different superclasses in the generated ontology. For example, GO-term 'vacuole' has superclass GO-term 'intracellular membrane-bounded organelle' and superclass GO-term 'cytoplasmic part'. From the generated ontology, we can easily observe the conceptual structure of the PubMed abstract collections. For instance, with respect to the first dataset, the documents allocated to GO-term 'microtubule cytoskeleton' are related to the documents allocated to GO-term 'actin cytoskeleton' because they share the same

Table 3
Comparison of GOClonto and other clustering algorithms in terms of key GO-term identification.

Dataset1	
GOClonto	Microtubule, microtubule cytoskeleton, gene expression, kinetochore microtubule, mrna processing, microtubule binding, cytokine production, actin cytoskeleton, translation, spindle microtubule, cytoplasmic microtubule, mitochondrial translation
STC	Cell, protein, study, resulting, show, activity, roles, complex function, pharmacology
Lingo	Unattached kinetochore, sex chromosome
Fuzzy Ants	Cells, cytoskeleton, region
Clustering based on TRS	Kinetochore, number change detected, genome was domesticated, aims congenital tufting, cytoskeleton, apoptotic signaling, lyophilized culture filtrate, talin, microtubules, mitotic exit pathway, caspase, upregulated, tumor
Dataset 2	
GOClonto	Membrane fusion, acrosome reaction, organelle, vesicle, intracellular organelle, mitochondrion, vacuole, lytic vacuole, lysosome, phagolysosome, acrosome, food vacuole
STC	Cells, protein, study, results, lysosome, membranes, vacuolization, function, suggesting, biosynthetic
Lingo	Sperm acrosome, mycobacterial phagosomes, parasitophorous vacuoles, boar spermatozoa, acrosome of spermatids, parasite and host, induction of autophagy
Fuzzy Ants	Cells, intracellular, essential
Clustering based on TRS	Stronger labeling, significant, proteins, sperm, mechanism-underlying the antihepatic, revealed that zip, Rpoe for growth, autophagic vacuoles, Thatcould, M.A. Ptb, cathepsin, procainamide, synthesis, immunodetection screen, human malaria

Table 4
Comparison of GOClonto and other clustering algorithms in terms of *F*-measure.

	GOClonto	STC	Lingo	Fuzzy Ants	Clustering based on TRS
Dataset1	**0.5294**	0.3884	0.3826	0.2203	0.1598
Dataset2	**0.6746**	0.5699	0.6610	0.4002	0.0874

superclass GO-term 'cytoskeleton'. Specifically, the documents allocated to GO-term 'cytoskeleton' also incorporates all documents allocated to its child GO-terms, including 'microtubule cytoskeleton' and 'actin cytoskeleton'. The generated ontology guarantees an 'is-a' relationship between GO-terms. The abstracts are thus sorted and categorized in an intuitive and semantically sound way. More surprisingly notable is that, in the GOClonto generated ontology the key GO-term 'organelle' appears as an ancestor to GO-terms 'microtubule cytoskeleton' and 'actin cytoskeleton' (Fig. 8). This is an example of how the ontology can discover some potential important key GO-terms which can not be extracted in the key Go-term induction phase. GO-term 'organelle' in this case is too general in the first PubMed abstract collection, but since its subclass GO-terms are recognized, the conceptual structure of the collections is still recreated quite accurately.

However, the hierarchical tree generated by the Fuzzy Ants algorithm does not maintain any clear relationship meaning. The relationships between terms are lexical rather than semantic. While some of the relationships are appropriate, e.g., 'cytoskeleton' is a subclass of 'cell', others are less meaningful, e.g., 'region' is a subclass of 'cytoskeleton' (Fig. 9). For the two document collections we used, the created hierarchical tree is meaningless for the purposes of conceptualization. We attribute this to the fact that GOClonto specifically aims at conceptualizing PubMed abstract collections, while the Fuzzy Ants clustering algorithm does not.

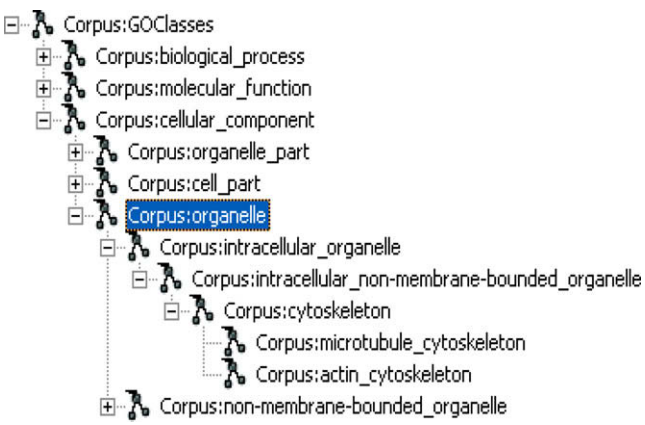

Fig. 8. A corpus-related gene ontology generated by GOClonto based on dataset1.

To conclude, the GOClonto method is able to identify the key GO-terms and generate corpus-related gene ontologies to represent the PubMed abstract collection. GOClonto has better performance than the other four clustering algorithms in terms of key GO-term identification and *F*-measure. We think this is because using LSA supports a more precise similarity calculation between GO-terms and PubMed abstracts. The ontology generated by GOClonto is more informative than the hierarchical tree created by the Fuzzy Ants clustering algorithm. This ontology can help users easily visualize the conceptual structure of the PubMed abstract collection and intuitively navigate its document groupings.

4. Conclusion and future work

In this paper, we proposed a novel method, GOClonto, which exploits GO to automatically generate corpus-related gene ontologies from PubMed abstract collections. The generated ontologies can help users conceptualize the PubMed abstract collections. Based on the vector space model, LSA techniques are used to identify the meaningful key GO-terms and allocated the related PubMed abstracts. By determining the superclass GO-terms of these GO-terms, ontologies are automatically generated and documents are linked to the generated ontologies through the key GO-terms. The experimental results show that GOClonto is able to identify key GO-terms from PubMed abstract collections and outperforms other four clustering algorithms. The generated ontologies are

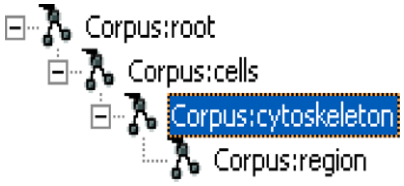

Fig. 9. A hierarchical tree created by the Fuzzy Ants clustering algorithm based on dataset1.

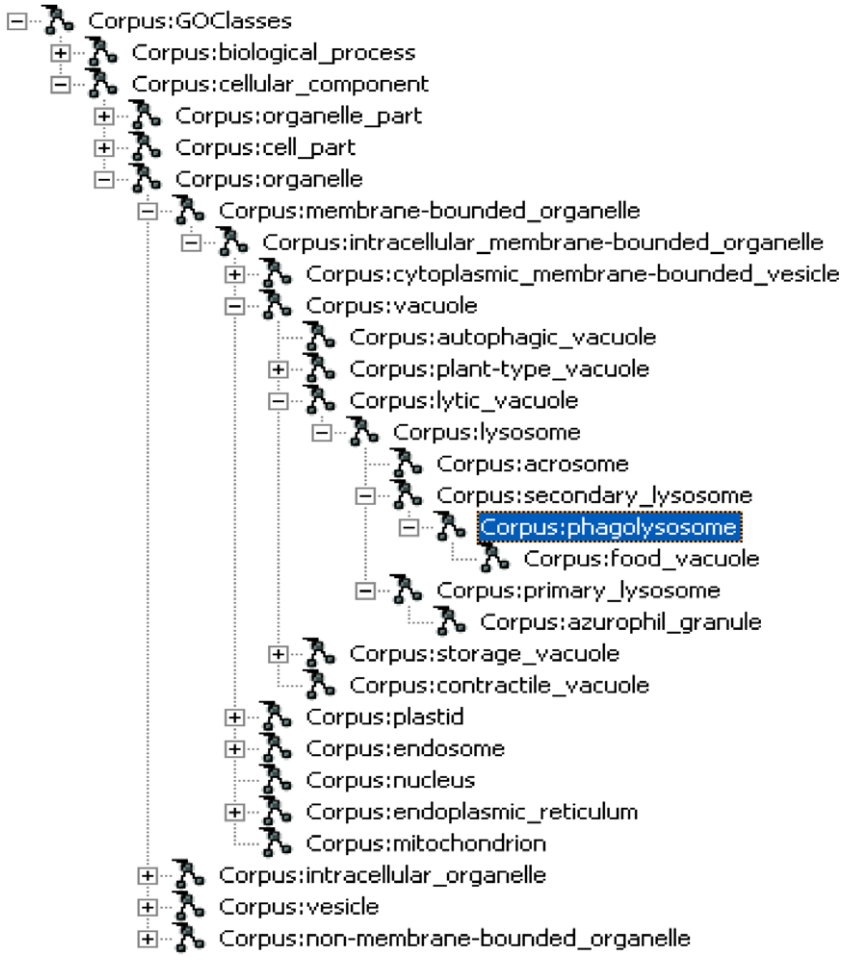

Fig. 10. A corpus-related gene ontology generated by GOClonto based on dataset2.

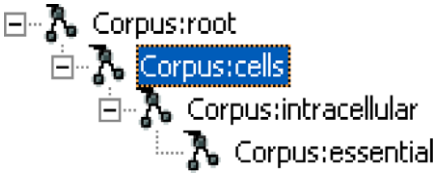

Fig. 11. A hierarchical tree created by the Fuzzy Ants clustering algorithm based on dataset2.

more informative than the hierarchical tree created by the Fuzzy Ants clustering algorithm. We believe that GOClonto will play an important role helping users visualize and conceptualize PubMed abstract collections.

One limitation of the GOClonto method is that its performance depends on the comprehensiveness of the ontology used. In particular, the GOClonto method can generate good ontologies related to document collections if the relations between concepts present in the collections are thoroughly represented in the existing ontology. In this paper, we used Gene Ontology as a domain-specific ontology, but for better results in other specific domains, more extensive ontologies should be incorporated.

We will conduct further research to improve our work in the following ways. First, we will incorporate more NLP methods to extract potential GO-terms from biomedical text. Second, addition of other visualization techniques alongside GOClonto can further aid user navigation of PubMed abstract collections. Third, we will study more ML (machine learning) algorithms to estimate the parameter values efficiently. Furthermore, we can consult with biomedical researchers and other professionals in order to gauge how best GOClonto can be used to support their work. Finally, other biomedical-related ontologies can be used to generate the ontologies. Good examples are FMA (the foundational model of anatomy) [34], which is known to be ontologically well designed, and SNOMED CT (systematized nomenclature of medicine—clinical terms) [35], which is a practical clinical ontology used by many hospitals.

Acknowledgment

This work was supported by ministry for health, welfare and family affairs, South Korea [A05-0909-A80405-06A2-15010A, Ontology-based EHR (electronic health record) Interoperability Technology Project].

References

[1] PubMed. Available from: http://www.ncbi.nlm.nih.gov/sites/entrez/.
[2] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Nat Genet 2000;25(1):25–9.
[3] Raychaudhuri S, Chang JT, Sutphin PD, Altman RB. Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. Genome Res 2002;12(1):203–14.
[4] Theodosiou T, Angelis L, Vakali A, Thomopoulos G. Gene functional annotation by statistical analysis of biomedical articles. Int J Med Inform 2007;76(8):601–13.

[5] Izumitani T, Taira H, Kazawa H, Maeda E. Assigning gene ontology categories (go) to yeast genes using text-based supervised learning methods. In: CSB'04: Proceedings of the 2004 IEEE computational systems bioinformatics conference. Washington, DC, USA: IEEE Computer Society; 2004. p. 503–4.
[6] Chen S-S, Kim H. Automated linking pubmed documents with GO terms using SVM. J Data Sci 2007;5(2):259–67.
[7] Vanteru B, Shaik J, Yeasin M. Semantically linking and browsing pubmed abstracts with gene ontology. BMC Genom 2008;9(Suppl 1):S10.
[8] Doms A, Schroeder M. Gopubmed: exploring pubmed with the gene ontology. Nucleic Acids Res 33 (web server issue).
[9] Delfs R, Doms A, Kozlenkov A, Schroeder M. Gopubmed: ontology-based literature search applied to gene ontology and pubmed. In: German bioinformatics conference. Bielefeld: Germany; 2004. p. 169–78.
[10] Smith TC, Cleary JG. Automatically linking medline abstracts to the gene ontology. In: ISMB 2003 BioLINK text data mining SIG; 2003.
[11] O. Zamir, O. Etzioni, Web document clustering: a feasibility demonstration. In: SIGIR'98: proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval; 1998. p. 46–54.
[12] Schockaert S. Het clusteren van zoekresultaten met behulp van vaagmieren (clustering of search results using fuzzy ants), Master thesis, University of Ghent.
[13] Lang NC. A tolerance rough set approach to clustering web search results, Master thesis, Warsaw University.
[14] Osinski S, Weiss D. A concept-driven algorithm for clustering search results. IEEE Intell Syst 2005;20(3):48–54.
[15] Zheng H-T, Kang B-Y, Kim H-G. Exploiting noun phrases and semantic relationships for text document clustering. Inf Sci 2009;179(13):2249–62.
[16] Iliopoulos I, Enright AJ, Ouzounis CA. Textquest: document clustering of medline abstracts for concept discovery in molecular biology; 2001. p. 384–95.
[17] Djebbari A, Karamycheva S, Howe E, Quackenbush J. Mesher: identifying biological concepts in microarray assays based on pubmed references and mesh terms. Bioinformatics 2005;21(15):3324–6.
[18] Yamamoto Y, Takagi T. Biomedical knowledge navigation by literature clustering. J Biomed Inform 2007;40(2):114–30.
[19] Homayouni R, Heinrich K, Wei L, Berry MW. Gene clustering by latent semantic indexing of medline abstracts. Bioinformatics 2005;21(1):104–15.
[20] Lin Y, Li W, Chen K, Liu Y. A document clustering and ranking system for exploring medline citations. J Am Med Inform Assoc 2007;14(5):651–61.
[21] Semantic_Relations. Available from: http://en.wiktionary.org/wiki/semantic_relation.
[22] OWL. Available from: http://www.w3.org/tr/owl-ref/.
[23] Stop_Word_List. Available from: http://www.lextek.com/manuals/onix/stopwords2.html.
[24] Phan X-H. Crftagger: Crf english pos tagger. Available from: http://crftagger.sourceforge.net/.
[25] Miller GA. Wordnet: a lexical database for english. Commun ACM 1995;38(11):39–41.
[26] Phan X-H. Crfchunker: Crf english phrase chunker. Available from: http://crfchunker.sourceforge.net/.
[27] Baeza-Yates RA, Ribeiro-Neto B. Modern information retrieval. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.; 1999.
[28] Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. J Am Soc Inform Sci 1990;41:391–407.
[29] Hill DP, Smith B, McAndrews-Hill MS, Blake JA. Gene ontology annotations: what they mean and where they come from? BMC Bioinform 2008;9(Suppl. 5):S2.
[30] Medical Subject Headings. Available from: http://www.nlm.nih.gov/mesh/.
[31] Steinbach M, Karypis G, Kumar V. A comparison of document clustering techniques. In: KDD workshop on text mining, 2000.
[32] Pantel P, Lin D. Document clustering with committees. In: SIGIR'02: proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA; 2002. p. 199–206.
[33] Weiss D. Descriptive clustering as a method for exploring text collections, Ph.D. thesis, Poznań University of Technology, Poznań, Poland; 2006.
[34] Rosse C, Mejino JLV. A reference ontology for biomedical informatics: the foundational model of anatomy. J Biomed Inform 2003;36(6):478–500.
[35] Stearns M, Price C, Spackman K, Wang A. Snomed clinical terms: overview of the development process and project status. Proc AMIA Symp 2001:662–6.