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