TaxoFinder: A Graph-Based Approach for Taxonomy Learning

Yong-Bin Kang, Pari Delir Haghigh, and Frada Burstein

Abstract—Taxonomy learning is an important task for knowledge acquisition, sharing, and classification as well as application development and utilization in various domains. To reduce human effort to build a taxonomy from scratch and improve the quality of the learned taxonomy, we propose a new taxonomy learning approach, named *TaxoFinder*. TaxoFinder takes three steps to automatically build a taxonomy. First, it identifies domain-specific concepts from a domain text corpus. Second, it builds a graph representing how such concepts are associated together based on their co-occurrences. As the key method in TaxoFinder, we propose a method for measuring associative strengths among the concepts, which quantify how strongly they are associated in the graph, using similarities between sentences and spatial distances between sentences. Lastly, TaxoFinder induces a taxonomy from the graph using a graph analytic algorithm. TaxoFinder aims to build a taxonomy in such a way that it maximizes the overall associative strengths among the concepts in the graph to build a taxonomy. We evaluate TaxoFinder using gold-standard evaluation on three different domains: *emergency management for mass gatherings*, *autism research*, and *disease* domains. In our evaluation, we compare TaxoFinder with a state-of-the-art subsumption method and show that TaxoFinder is an effective approach significantly outperforming the subsumption method.

Index Terms—Taxonomy learning, ontology learning, concept taxonomy, concept graph

1 Introduction

Taxonomies are the key to developing successful applications in a domain, such as information retrieval (IR), knowledge searching and classification [1], [2]. In particular, considering the ever-growing amount of text digital data per year, *taxonomy learning from text* is a primary research area for developing such applications nowadays [2]. In a given domain, the goal of taxonomy learning is to automatically or semi-automatically build a taxonomy by identifying *domain-specific concepts* (hereafter, we refer to it as *concepts*) and their *taxonomic relations* from the domain text corpus, with other relevant knowledge if it is available.

An important nature of a taxonomy is that it enables representing highly related concepts together, and the path between two concepts that reflect how these are semantically related in the domain. A taxonomy is often referred to as the 'backbone of an ontology' built using the most important 'is-a' relationship [2]. Due to this link, taxonomy learning is sometimes regarded as the prerequisite step for *ontology learning* that aims to extract concepts, relations and occasional axioms about the concepts to build an ontology [2].

Manually building a taxonomy poses a great challenge that requires a huge amount of time and effort of humans. Taxonomy learning uses methods developed in the fields of natural language processing (NLP), information retrieval and machine learning (ML) in an attempt to reduce the human effort and build a high quality taxonomy [2]. Most existing approaches use pattern-based [3], [4], clustering [5], [6], statistical [2], [7], and graph-based approaches [8], [9] to build a taxonomy from extracted concepts.

However, the nature of 'context' surrounding the concepts and its actual impact have little been studied for taxonomy learning. This motivates us to thoroughly analyze the statistical and semantic relationships between the concepts, considering their contexts, to build a taxonomy. As a context of a given concept, in this work, we use the *sentence* encompassing the concept, as a sentence is generally seen as a linguistic unit consisting of words that are meaningfully linked together.

This paper proposes a graph-based unsupervised approach, named *TaxoFinder*, for taxonomy learning that automatically builds a taxonomy from a *semantic graph*, named *CGraph*, of concepts modelled from a target corpus. First, we extract concepts from the corpus using a concept extractor [10]. Second, based on the *co-occurrences* of the concepts in a *sliding window*, which is the set of consecutive (or sequential) sentences in each document from the corpus, we build an undirected graph, CGraph. In the CGraph, a node is a concept and an edge is created if two concepts co-occur in a sliding window thus making an association between them. From the CGraph, we measure the *associative strength* between two concepts by leveraging the sentence information that the concepts appear in the corpus. Lastly, we induce a taxonomy from the CGraph by applying a *maximum spanning tree* (MST) algorithm [11].

This paper makes two main contributions: First, we propose to build a CGraph reflecting semantic associations and associative strengths of concepts from a text corpus. Using an MST algorithm, we show how to induce a taxonomy from the graph. Second, we propose a method for combining the following three knowledge components to quantify the associative strengths between two concepts in a CGraph:

• *The authors are with Faculty of Information Technology, Monash University, Melbourne, Australia.*
E-mail: {yongbin.kang, pari.delir.haghighi, frada.burstein}@monash.edu.

Manuscript received 14 Jan. 2015; revised 28 July 2015; accepted 24 Aug. 2015. Date of publication 2 Sept. 2015; date of current version 6 Jan. 2016. Recommended for acceptance by G. Li.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2015.2475759

(1) their *co-occurrences* in the corpus, (2) the *spatial proximity* (i.e. *distance*) and (3) *semantic similarity* between the sentences where those concepts appear in the corpus. The intuition behind is that incorporating such sentence information can be useful to measure the influences of contexts surrounding the concepts for measuring the associative strengths among concepts.

We evaluate TaxoFinder using *gold-standard evaluation* [12] on three domains: *emergency management for mass gatherings*, *autism research domain*, and *disease domain*. In each domain, we evaluate how TaxoFinder effectively builds taxonomic relations between concepts closer to the gold-standard taxonomy. To determine the quality of TaxoFinder, we compare it with a state-of-the-art *subsumption* method (SSM) [1]. Our evaluation shows that TaxoFinder significantly outperforms the subsumption method on all three domains.

This paper is organized as follows. Section 2 reviews current major approaches to taxonomy learning and highlights main features of TaxoFinder. Section 3 presents the details of TaxoFinder. Section 4 describes the evaluation of TaxoFinder, and Section 5 concludes the paper and presents future work.

2 Related Work

Over the past decade, different approaches have been studied for taxonomy learning using various techniques such as natural language processing, information retrieval, machine learning and data mining (DM) [2]. NLP provides tools for finding concepts and their taxonomic relations based on *linguistic patterns* and their *semantic relations*, derived from a lexical database (e.g., WordNet[1]). IR provides techniques for analysing taxonomic relations of concepts often based on their *similarity* or *co-occurrences*. ML and DM techniques contribute to finding *interesting patterns* between concepts observed in the given corpus and learning from such patterns to infer taxonomic relations for unknown concepts. We classify the existing approaches to taxonomy learning from text into the following four categories.

2.1 Taxonomy Learning Approaches

First, *pattern-based approaches* typically apply NLP and ML techniques, and are based on predefined *lexico-syntactic patterns* (e.g., 'NP such as NP') to extract concepts and their taxonomic relations. One of the pioneering works is proposed in [3] where some lexico-syntactic patterns (e.g., '*is-a*') are manually identified and then more patterns are learned from a bootstrapping algorithm.

Large corpora (i.e. the web) were also used in a bootstrapping algorithm as an attempt to iteratively learn more patterns [4]. In [13], a combination of deep linguistic patterns and ML algorithms (e.g., support vector machine) was used to infer concept relations from text. To enlarge the coverage of concept relations, [14] investigated methods for leveraging a large semantic network such as Wikipedia with lexico-syntactic patterns. With lexico-syntactic patterns, *verbal-noun dependency relations* between concepts extracted from text were also used to derive their taxonomic relations [15].

1. http://wordnet.princeton.edu

Second, *clustering approaches* formulate the problem of taxonomy leaning as a clustering or classification problem, and often achieved based on IR and ML. These approaches assign concepts into unknown *clusters* in such a way that concepts in the same cluster are similar to each other in some sense than those in other clusters [16]. Particularly, *hierarchical clustering* techniques were popularly studied [5], [6]. Often, these were based on the idea that concepts are merged on *a series of levels* in which all concepts are initially considered to be individual clusters, and these are subsequently merged into larger clusters until all concepts form one cluster. Clusters of concepts were formed based on concept similarity derived from WordNet, the domain corpus or the Web. Then, these clusters were used to determine taxonomic relations between concepts.

Third, *statistical approaches* typically use IR, ML and DM, and are often based on the premise that concepts, semantically related, tend to be near or co-occur together in a document [2]. The more two concepts co-occur in a context (e.g., a sentence, document), the more semantically related these are. An approach based on *high-order (or indirect order) occurrences* between concepts for inferring their taxonomic relations was proposed [17]. In another study, applying the *probabilistic topic models* for taxonomy learning was introduced [7]. Given a set of concepts, this approach proposed that Information Theory [18] can be used as a probabilistic proxy for learning taxonomic relations between concepts. The *subsumption method* has been widely studied, focuses on the co-occurrences of concepts [1], [19]. The idea is that a concept A subsumes another concept B (i.e. A is the hypernym of B) if the documents (or some proportion of the corpus) that B appears are a subset of the documents that A appears.

Fourth,*graph-based approaches* are commonly based on the scheme that builds a *graph* in which nodes represent concepts and edges their taxonomic relations. The distance of each edge approximates the relationship strength between two concepts. From the graph, a taxonomy can be finally inferred using heuristics. A graph-based approach was introduced in [8] that leverages the Web to build a taxonomy from a directed graph. Given a root and basic level concepts, it finds their new possible hypernym/hypopnym concepts (called intermediate concepts) using predefined lexico-syntactic patterns, until no new concepts are found in order to derive a taxonomy. Another study, OntoLearn Reloaded [9], finds *definition sentences* for each concept. Such sentences are identified by harvesting all sentences that contain the given concept in the corpus and the web. It then uses the results of a classifier to build a directed graph with the concepts as nodes and the relations as edges. From this complex graph, a taxonomy is finally induced through heuristics considering incoming/outgoing edges, path-length and connectivity of nodes.

Considering the above mentioned four categories, our work falls under the graph-based approaches.

2.2 TaxoFinder versus Existing Approaches

TaxoFinder differs from the above approaches in the following aspects: Unlike the *pattern-based approaches*, TaxoFinder does not rely on predefined lexico-syntactic patterns that often require additional investigation on extra knowledge sources or classifiers to learn more patterns to discover

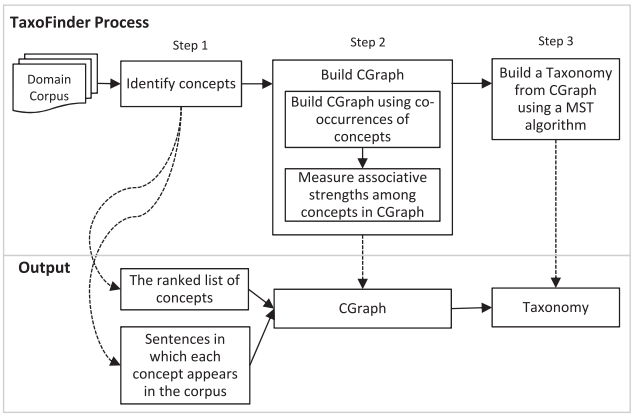


Fig. 1. The overview of TaxoFinder.

taxonomic relations of concepts. Compared to the *clustering* and *statistical approaches*, TaxoFinder builds a graph which characterizes associations between concepts, and then induces a concept taxonomy from the graph with a graph analytic algorithm. Moreover, our metric for measuring such associative strengths is new and distinguished from the similarity metrics used in the approaches in that TaxoFinder considers and combines co-occurrence of concepts, the distance between sentences where the concepts appear together, and the semantic similarity between those sentences. TaxoFinder differs from the existing *graph-based approaches* in that it measures the associative strength between concepts using the combination of the above three factors, unlike [8], [9] that determine taxonomic relations using predefined lexico-syntactic patterns.

3 Building Concept Taxonomy

TaxoFinder performs three steps to derive a taxonomy as outlined in Fig. 1. First, it identifies concepts from the domain corpus using a concept extractor. The output of this step includes (1) the ranked list of concepts according to their domain relevance and (2) sentence information in which each concept appears in each document in the corpus. Second, TaxoFinder builds a CGraph which represents how concepts are associated together based on their co-occurrences. Using the sentence similarity and sentence distance measures, their associative strength is quantified. Lastly, a taxonomy is built from the CGraph using a graph analytic algorithm. In the following, we describe each of these steps in more detail.

3.1 Identifying Concepts from a Domain Corpus

Given a domain corpus, *concept extraction* is the first step for taxonomy learning [10]. If extracted concepts are irrelevant, a taxonomy may not correctly represent domain knowledge as such irrelevant concepts can also lead to generating irrelevant taxonomic relations.

Most existing approaches to concept extraction can be classified into the four categories [10]: (1) *Machine learning approaches* that identify concept candidates from a corpus using NLP techniques and then learn a classifier to identify which candidates are most likely to be concepts; (2) *Multiple corpus-based approaches* that first identify concept candidates using NLP techniques and then use statistical distribution of them across multiple corpora of different domains to identify concepts; (3) *Glossary-based approaches* that make use of author-provided glossary terms in a corpus to identify key concepts. (4) *Heuristic-based approaches* which depend largely on different weighting schemes of noun phrases (e.g., variant forms of TF-IDF, statistical distribution of noun phrases), occurrence position of phrases, and/or phrase length in words. Recently, a state-of-the-art heuristic-based approach, named CFinder [10], was introduced that combines linguistic patterns, statistical distribution, domain-specific characteristics and inner structural pattern of extracted terms, and its extensive evaluation showed its effectiveness over other approaches.

In addition to the above four categories, MetaMap[2] [20] is a tool for extracting biomedical concepts from input text through investigating their semantic relationships found in *Unified Medical Language System (UMLS)* Metathesaurus [21].

As the output of the concept extraction step, we obtain the ranked list of concepts according to their domain relevance. In addition, we obtain a set of sentences in which each concept appears in each document in the corpus. These information sources will be used to build a CGraph. TaxoFinder can be incorporated with any concept extraction method which can generate such information sources.

3.2 Building a CGraph Using Extracted Concepts

This section discusses the second step of TaxoFinder. Using extracted concepts and the sentences in which these concepts appear generated in Section 3.1, we build a CGraph where a node represents each of such concepts and an edge represents an association between nodes. Each edge has a weight indicating the *associative strength* between two nodes. We now describe the sub-steps of this second phase for building a CGraph to learn a taxonomy.

3.2.1 Symbolic Notations

We first present basic notations that will be used throughout the paper:

- Let $\mathbf{D}$ be a corpus in the target domain which is a collection of k text articles: $\mathbf{D} = \{D_1, \ldots, D_k\}$.
- Let $\mathbf{S}$ be the set of all sentences that appear in all documents in $\mathbf{D}$.
- Let $\mathbf{C}$ be the set of all concepts extracted from $\mathbf{D}$.
- Let $D_j \in \mathbf{D}$ consist a sequence of m sentences that appear in D_j, and be represented as: $D_j = \langle s_{j1}, \ldots, s_{jm} \rangle$.
- Let $s_{ji} \in \mathbf{S}$ be the ith sentence in D_j and be represented as a set of n concepts that appear in the sentence.
- Let $s_j(c_k) \in \mathbf{S}$ be the set of sentences that contain a concept $c_k \in \mathbf{C}$ that appear in $D_j \in \mathbf{D}$.
- Let $I_j(c_k)$ be the set of sequential indices of sentences in $s_j(c_k)$.

3.2.2 Building a Bigraph for CGraph

A CGraph is built from concepts joined by undirected edges. This graph is initially built from a special *bipartite*

2. http://metamap.nlm.nih.gov/

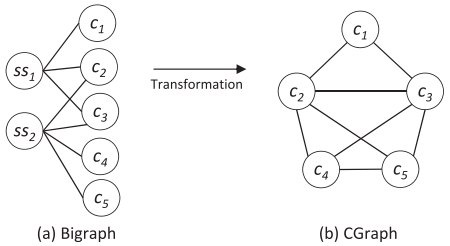


Fig. 2. A bigraph is transformed into a CGraph.

graph (simply *bigraph*) G that consists of two kinds of nodes, one representing *concepts* in **C** and the other representing *the collection of sets of sequential sentences* in **S** that contain the concepts. In G, the concepts are not connected to any other concepts, and the same is valid for the sets of sequential sentences.

In G, an edge, between a concept and a set of sequential sentences, represents that the concept appears in the set. From G, we construct a CGraph in which two concepts are connected if they appear together in the same set of sequential sentences. Such sequential sentences are referred to only those sentences that sequentially appear in each document $D_j \in \mathbf{D}$. As a unit of the set of sequential sentences, we use a *sliding window*. For example, suppose we set the size of the sliding window to be 11. Then, for each sentence $s_i \in \mathbf{S}$ in D_j, its sliding window is formed as its five preceding sentences $\{s_{i-5}, \ldots, s_{i-1}\}$, s_i and its five following sentences $\{s_{i+1}, \ldots, s_{i+5}\}$ in D_j.

An example of how to build a CGraph from G is shown in Fig. 2: Suppose that ss_1 and ss_2 are two sets of sequential sentences in D_j, where these sets are built using the same sliding window size. Also, suppose that $\{c_1, \ldots, c_5\}$ are some of the concepts extracted from D_j. The left bigraph G can be transformed into the right undirected graph CGraph, where three concepts $\{c_1, c_2, c_3\}$ are connected to each other as they appear together in the set ss_1. For the same reason, four concepts $\{c_2, c_3, c_4, c_5\}$ are connected to each other as they all appear in the set ss_2. Note that, however, a concept c_1 is not connected to two concepts $\{c_4, c_5\}$ as c_1 does not appear in ss_2 where c_4 and c_5 appear in.

In this step, our intuition is that if two concepts appear in the same set of sequential sentences, they are semantically associated to each other in the domain. This intuition has also been effective to automatically learn interesting relationships between terms in a text corpus on different domains [22], [23].

3.2.3 Measuring Associative Strengths of Concepts

A key challenge in constructing the CGraph from concepts in **C** is the calculation of an *associative strength* between two concepts. This strength quantifies how semantically close these two concepts are. The closer two concepts appear together, the stronger their associative strength is considered. The associative strengths among all extracted concepts will be used as the key for building a taxonomy from the CGraph which will be discussed as the third step in Section 3.3.

We define the associative strength between two concepts c_1 and c_2, denoted by $w(c_1, c_2)$, with respect to the corpus **D** as follow:

$$w(c_1, c_2) = \frac{1}{k} \sum_{j=1}^{k} w_j(c_1, c_2), \tag{1}$$

where k is the number of documents in **D** (i.e. $k = |\mathbf{D}|$), and $w_j(c_1, c_2)$ represents the associative strength between c_1 and c_2 with respect to a document $D_j \in \mathbf{D}$. Thus, $w(c_1, c_2)$ is calculated as the mean of the associative strengths between c_1 and c_2 across all documents in **D**. The value of $w(c_1, c_2)$ is normalized between 0 and 1, where 1 means that the associative strength between two concepts is the highest, and 0 indicates that the strength is lowest.

To define $w_j(c_1, c_2)$, we use a combination of two intuitions: (1) the more semantically similar c_1 and c_2 are, the stronger their associative strength is, and (2) the closer two concepts appear in a document, the stronger their associative strength is. To reflect the first intuition, we calculate semantic similarity between two concepts.

The importance of the notion of *context* has been emphasized in many studies in information retrieval [24]. Measuring similarity between two concepts (or terms) without considering their contexts in most cases can reduce the confidence of results. This is also in accord with 'distributional hypothesis' that similar words tend to appear in similar contexts [25]. Thus, we consider a context to measure semantic similarity between two concepts. As the 'context' of a concept in a document, we use the 'sentence' where the concept appears in.

Therefore, to implement the first intuition, given two concepts, we consider the similarities of all pairs of sentences where each concept appears in. To implement the second intuition, we also consider the distance of the contexts (i.e. sentences) where the two concepts appear in. These two intuitions are combined in a unified formula which will be discussed in the later part of this section.

Formally, given two concepts c_1 and c_2 with respect to a document $D_j \in \mathbf{D}$, we define the function $w_j(c_1, c_2)$ in (1) as follows:

$$w_j(c_1, c_2) = \frac{1}{m * n} \sum_{p,q} as(s_{jp}, s_{jq}), \tag{2}$$

where:

- $as(s_{jp}, s_{jq})$ represents the function that calculates the associative strength between two sentences s_{jp} and s_{jq} in $D_j \in \mathbf{D}$, where $s_{jp} \in s_j(c_1)$ and $s_{jq} \in s_j(c_2)$;
- p and q are the sentence sequential indices belonging to $I_j(c_1)$ and $I_j(c_2)$ in D_j, respectively, i.e., $p \in I_j(c_1)$ and $q \in I_j(c_2)$;
- $I_j(c_k)$ where $k \in \{1, 2\}$ represents the set of sequential indices of the sentences in $s_j(c_k)$, i.e. the set of sentences that contain c_k that appears in D_j, according to the definition in Section 3.2.1. Formally, $I_j(c_k)$ is formed using the following definition:

$$I_j(c_k) = \{i \mid \text{if } c_k \text{ appears in } s_{ji} \in D_j\}, \tag{3}$$

  where $i \in [1, |D_j|]$ and $D_j = \langle s_{j1}, \ldots, s_{j|D_j|} \rangle$; and
- m and n are the number of elements in $I_j(c_1)$ and $I_j(c_2)$, respectively, i.e., $m = |I_j(c_1)|$, $n = |I_j(c_2)|$.

Formula (2) calculates the associative strength between c_1 and c_2 using the associative strengths among their

contexts; more specifically, the mean of the associative strengths between all pairs of two sentences, $s_j(c_1)$ and $s_j(c_2)$, that contain c_1 and c_2 in D_j, respectively. Therefore, if the associative strength between $s_j(c_1)$ and $s_j(c_2)$ is stronger, we also derive that the associative strength between c_1 and c_2 is stronger.

Now, our challenge is how to define the function $as(s_{jp}, s_{jq})$ in Formula (2) such that it satisfies the two intuitions presented above. For this, our approach considers two observations that have been often found in general text articles: (1) the closer two sentences are in a document, the more semantically associated they are considered (i.e. linguistic cohesion) [23], and (2) the more semantically similar two sentences are, the more strongly associated they are [26]. Based on these two observations, our underlying premise for defining $as(s_{jp}, s_{jq})$ is that (1) the associative strength between two sentences is stronger, if these are semantically more similar to each other (i.e. sentence similarity), and (2) if these are closer to each other (i.e. sentence distance).

Formally, we define $as(s_{jp}, s_{jq})$ in Formula (2) as follows:

$$as(s_{jp}, s_{jq}) = sim(s_{jp}, s_{jq})^{|p-q|}, \tag{4}$$

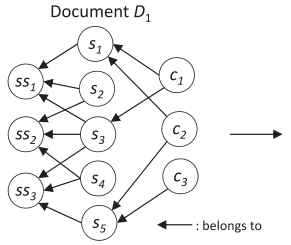
where the sentence similarity $sim(s_{jp}, s_{jq})$, between two sentences s_{jp} and s_{jq}, is based on the approach proposed in [27] which showed a high performance in [26]:

$$sim(s_{jp}, s_{jq}) = \frac{\sum_{w \in s_{jp}} sim_{\max}(w, s_{jq}) + \sum_{w \in s_{jq}} sim_{\max}(w, s_{jp})}{|s_{jp}| + |s_{jq}|}, \tag{5}$$

where $sim_{\max}(w, s_{jq})$ is the highest semantic similarity score between $w \in s_{jp}$ and words having the same Part-Of-Speech (POS) tag as w in s_{jq}. To measure the similarity between two words in (5), we use the following word-to-word similarity measures: First, for words with the same POS tag (i.e. noun, verb, adjective or adverb), we use a well-known word-to-word semantic similarity method using WordNet [28]. WordNet can be seen as a lexical ontology where concepts correspond to *word senses*, and concept labels are denoted as *synsets*—groups of synonym words (each synset expresses a distinct concept). To measure the similarity, the correct sense of each word compared needs to be determined (i.e. *word sense disambiguation (WSD)* [29]). For this, we use the *first* sense of each word provided in WordNet as the senses of words in WordNet are ranked according to frequency. Thus, our premise is that the first sense is the most important, representative sense of a given word. Second, for words having the other POS tags, we use instead a lexical match measure (i.e. the edit distance) that assigns 1 to $sim_{\max}$, if the words are lexically identical.

The similarity in Formula (5) is normalized between 0 and 1, where 1 indicating identical sentences, and 0 no semantic overlap between the two sentences.

As defined in Formula (4), we combine the sentence similarity Formula (5) with the distance $|p-q|$ of the two sentences s_{jp} and s_{jq}. The closer the indices of two sentences is, the stronger their associative strength is. The difference $|p-q|$ is used as a scaling exponent of the sentence similarity. Thus, the exponent $|p-q|$ is used to exponentially penalize the similarity value further if the two sentences are more distant.



(a) Document D_1

sentence similarities
$sim(s_{11}, s_{13})$ = 0.6
$sim(s_{11}, s_{15})$ = 0.5
$sim(s_{13}, s_{15})$ = 0.7

(b) Sentence similarity

associative strength
$w(c_1, c_2)$ = 0.48
$w(c_1, c_3)$ = 0.28
$w(c_2, c_3)$ = 0.53

(c) Associative strength

Fig. 3. An example of measuring the associative strengths.

To sum up, referring to Formulas (1)-(5), the calculation of the associative strength between two concepts is sensitive to both the *semantic* information shared by the sentences that contain the concepts and the *distance* between the sentences. If two concepts appear in two sentences whose semantic similarity is higher, their associative strength is closer to 1. This associative strength is exponentially penalized by the distance between the two sentences.

To illustrate how to calculate the associative strength between two concepts, let us consider an example in Fig. 3. Suppose that there is a corpus **D** which has a document D_1 consisting of five sentences, $D_1 = \{s_1, s_2, s_3, s_4, s_5\}$. Suppose that ss_1, ss_2 and ss_3 are three sentence sets, each having k-sequential sentences that appear in **D**. Assuming that a sliding window size is 3 (i.e. $k = 3$), we set $ss_1 = \{s_1, s_2, s_3\}$, $ss_1 = \{s_2, s_3, s_4\}$, and $ss_3 = \{s_3, s_4, s_5\}$. In Fig. 3a, each directed edge from one a to the other b means a belongs to b. For example, looking at ss_1, s_1 and c_1, we see that s_1 belongs to ss_1, and c_1 belongs to (i.e. appears in) s_1.

Suppose that we also extracted three concepts c_1, c_2, and c_3 from D_1, i.e., $\mathbf{C} = \{c_1, c_2, c_3\}$, where c_1 appears in s_1 and s_3, c_2 appears in s_1 and s_5, and c_3 in s_5. Thus, s_{11} is the sentence represented as $s_{11} = \{c_1, c_2\}$, since these two concepts appear in the sentence s_1. Also, $s_{13} = \{c_1\}$, $s_{15} = \{c_2, c_3\}$. In addition, $s_1(c_1)$ denotes the set of sentences that contain the concept c_1 appears in D_1, i.e. $s_1(c_1) = \{s_1, s_3\}$. Also, $s_1(c_2) = \{s_1, s_5\}$, and $s_1(c_3) = \{s_5\}$. Also, $I_1(c_1)$ indicates the set of sequential indices of sentences $s_1(c_1)$, thus $I_1(c_1) = \{1, 3\}$. Also, $I_1(c_2) = \{1, 5\}$, and $I_1(c_3) = \{5\}$.

Using the notations in Section 3.2.1 and assuming the sentence similarities as Fig. 3b, the associative strength between two concepts c_1 and c_2 with respect to **D**, $w(c_1, c_2)$, is calculated as (see also Fig. 3c):

$$\begin{aligned} w(c_1, c_2) &= w_1(c_1, c_2) \\ &= \frac{as(s_{11}, s_{11}) + as(s_{11}, s_{15}) + as(s_{13}, s_{11}) + as(s_{13}, s_{15})}{4} \\ &= \frac{1^0 + 0.5^4 + 0.6^2 + 0.7^2}{4} = 0.48. \end{aligned}$$

Following the above calculation, we can also obtain $w(c_2, c_3)$ as 0.28 and $w(c_1, c_3)$ as 0.53.

3.3 Deriving a Taxonomy from CGraph

Once we build a CGraph, the third step is to derive a taxonomy from it. Our eventual goal is to build a taxonomy in such a way that it *maximizes the overall associative strengths among all concepts in CGraph* to find a good taxonomy. This is aligned with the notion of a good taxonomy used in the prior work [30]. The learned taxonomy guarantees that highly associated concepts are closely positioned.

The more concepts a CGraph has, the more complicated the CGraph tends to be, as the number of edges in the CGraph could be substantially increased as a result. Note that the maximum number of edges in a CGraph is $|\mathbf{C}| * (|\mathbf{C}| - 1)/2$, where $|\mathbf{C}|$ is the number of nodes in the graph. From a CGraph, one possible way for deriving a taxonomy might be to reduce the overall number of edges in the graph by adjusting the size of the sliding window. If we reduce this size, the number of concepts that co-occur together will be decreased, and thus the number of associations among the concepts will be also decreased. Another possible way might be removing edges with very low associative strengths. However, both ideas cannot guarantee that the reduced graph would form a taxonomy, where the total number of edges must be $(|\mathbf{C}| - 1)$. Although these ideas might be able to build $(|\mathbf{C}| - 1)$ edges, they will not always result in forming a taxonomy (i.e. not a hierarchical structure). Also, these ideas do not consider the connectivity in the resulting taxonomy thus easily producing disconnected graphs.

In TaxoFinder, given the concept set $\mathbf{C}$, we construct a *spanning tree* to derive a taxonomy by reducing a substantial amount of the edges from the CGraph. This approach guarantees that the learned taxonomy has $(|\mathbf{C}| - 1)$ edges and it connects all the nodes in $\mathbf{C}$. A spanning tree is a minimal set of edges that connect all nodes in a graph. In particular, we build a *maximum spanning tree* that connects all the nodes in CGraph with the *maximum sum* of the associative strengths between them and discards the edges with less significant associative strengths. This spanning tree is our learned taxonomy that eventually captures the maximum associative strengths among the concepts in the CGraph.

Formally, given a CGraph with nodes (or concepts) $\mathbf{C}$, we construct a taxonomy, which is a subgraph T, defined as follows:

$$\underset{T \subseteq \text{CGraph}}{\operatorname{argmax}} \sum w(c_i, c_j), \tag{6}$$

for all $(c_i, c_j) \in T$ and $w(c_i, c_j)$ is the associative strength between c_i and c_j calculated by Formula (1). Note that we must decide which node (concept) should be the root of T. Intuitively, the root node is the most relevant concept in the target domain among all the concepts in T. Given a ranked list of concepts extracted by a concept extractor, the first ranked concept is the most relevant concept. Thus, we choose it as the root node in T.

To build a taxonomy, initially T has the first ranked concept and is repeatedly augmented with the edge (x, y) with the maximum associative strength such that $x \in T, y \notin T$ from the CGraph using the Prim's algorithm [31]. An example of building a taxonomy is shown in Fig. 4. The label on each edge indicates the associative strength between the two connected nodes. Initially, the algorithm starts with c_1 (the root concept) and subsequently chooses the edges (c_1, c_2), (c_2, c_5), (c_5, c_3) and (c_5, c_4) to build a taxonomy. Note that a path in the resulting MST is only made if there is an association in the given CGraph.

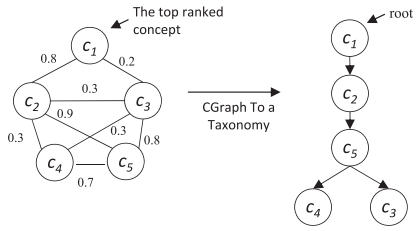


Fig. 4. The derivation of a taxonomy from a CGraph.

Eventually, using the MST algorithm on the CGraph, we accomplish our goal—the constructed taxonomy maximizes the overall associative strengths among all concepts $\mathbf{C}$ with $(|\mathbf{C}| - 1)$ edges from the CGraph and enables highly associated nodes to be closely positioned together.

4 Evaluation

Evaluating a learned taxonomy is a crucial task for assessing its quality and ensuring that it is the optimal representation of a domain. In general, ontology (or taxonomy) evaluation[3] is known to be a complicated task and even for humans as there is no clear, unique way for modeling domain knowledge [9]. In this section, we first review existing evaluation approaches. Then, we will examine evaluation criteria for one of the widely-used approaches, *gold-standard evaluation* in more detail, which is also used in this work. After that we present our evaluation method, and finally discuss evaluation results.

4.1 Taxonomy Evaluation Approaches

Existing approaches to evaluating learned taxonomies can be divided into four categories based on how evaluation is made [12]: (1) *Application-based (or task-based) evaluation* evaluates the quality of learned taxonomies in the context of applications by measuring their impact on the aspect of improving the performance of applied applications [32], [33]; (2) *Data-driven (or corpus-based) evaluation* evaluates the fitness between a learned taxonomy and a domain-specific corpus representing the knowledge of the target domain [34], [35]. It usually measures the terminological coverage of the learned taxonomy with respect to extracted key terms from the corpus; (3) *Domain-expert evaluation* relies on human judges with relevant domain expertise to assess the quality of learned taxonomies [9], [19]; and (4) *Gold-standard (or reference-standard evaluation)* is the most popular approach for taxonomy evaluation and compares a learned taxonomy with a gold-standard taxonomy [1], [9], [15]. This approach assumes that a gold-standard taxonomy is available in the target domain and both the learned and gold-standard taxonomies are built using the same set of terminological concepts [36].

We now more specifically examine evaluation metrics used in 'gold-standard evaluation'. The metrics are generally

3. In general, taxonomy evaluation is part of ontology evaluation as taxonomic relations are the representative relations in ontologies.

divided into a *local* and a *global* measures [12]. The local measure is used for comparing the position of a concept in the learned taxonomy T_l with that of the same concept in the gold-standard taxonomy T_g. Thus, the concepts compared must exist in both taxonomies. The global measure then computes the local measures of all concepts in T_l, thus providing the overall taxonomic quality of T_l.

To define a local measure, the notion of the *common sematic cotopy (csc)* is often used [1], [6]. The csc represents the collection of a concept and its both super-concepts and sub-concepts shared by both T_l and T_g. Formally, given a concept $c \in T_l$ with respect to T_g, its csc is defined as [1]:

$$csc(c, T_l, T_g) = \{c_i | c_i \in C_l \cap C_g \ (c_i \leq_{C_l} c \vee c \leq_{C_l} c_i)\}, \quad (7)$$

where C_l is the set of concepts in T_l, C_g is the set of concepts in T_g, and '$\leq_{C_l}$' is the order induced by taxonomic relations in T_l (i.e. c_i is a either a sub-concept of c ($c_i < c$) or super-concept ($c < c_i$) or the equivalent of c ($c_i = c$)).

Based on this notion, *local taxonomic precision (tp)* and *local taxonomic recall (tr)* are defined using the similarity of concepts' positions in T_l and T_g, which measure the quality of the learned relations of each concept $c \in T_l$, denoted as [1]:

$$\begin{aligned} tp(c) &= \frac{|csc(c, T_l, T_g) \cap csc(c, T_g, T_l)|}{|csc(c, T_l, T_g)|}, \\ tr(c) &= \frac{|csc(c, T_l, T_g) \cap csc(c, T_g, T_l)|}{|csc(c, T_g, T_l)|}, \\ tf(c) &= 2 \cdot tp(c) \cdot tr(c)/(tp(c) + tr(c)), \end{aligned} \quad (8)$$

where $tf(c)$ is *local taxonomic F-measure* of c, calculated as a harmonic mean of $tp(c)$ and $tr(c)$. Then, *global taxonomic precision* (TP), *global taxonomic recall* (TR), and *global taxonomic F-measure* (TF) are defined to measure the quality of the relations between two taxonomies T_l and T_g [1]:

$$\begin{aligned} \mathrm{TP} &= \frac{1}{|C_l \cap C_g|} \sum_{c \in C_l \cap C_g} tp(c), \\ \mathrm{TR} &= \frac{1}{|C_l \cap C_g|} \sum_{c \in C_l \cap C_g} tr(c), \\ \mathrm{TF} &= 2 \cdot \mathrm{TP} \cdot \mathrm{TR}/(\mathrm{TP} + \mathrm{TR}), \end{aligned} \quad (9)$$

where the higher a TF value is, the better the quality of the learned taxonomy is.

4.2 Evaluation Framework

To determine whether TaxoFinder is an effective method for taxonomy learning, we evaluate it across three domains: (1) *emergency management for mass gatherings* (simply, EMD) [37] in which our aim is to extract concepts for medical emergency management for mass gatherings and build a taxonomy using them, (2) *autism research domain* (simply, ARD) in which our aim is to come up with concepts specific to autism research and build their taxonomic relations, thus providing a valuable insight into the nature of autism research, and (3) *disease domain* (simply, DD) where our focus is to extract concepts related to general *disease* and build a taxonomy using them.

As a gold-standard taxonomy for evaluation, we used an ontology DO4MG [37] for EMD which has been recently developed via a thorough evaluation by domain experts; for ARD, we selected the recently-developed autism phenotype (ASDPTO) ontology [38]. Although it focuses more on conceptualizing knowledge about the Autism Spectrum Disorder behavioral phenotype, it represents part of the knowledge within ARD and could be considered as a gold standard taxonomy; and for DD, we used a disease taxonomy within *(Medical Subject Headings)* MeSH[4] that is a representative, biomedical controlled vocabulary consisting of 26k+ biomedical terms arranged in a taxonomic structure introduced by National Library of Medicine (NLM).

TABLE 1
A Summary of the Experimented Corpus

Domain	Doc#	Total Sentence#	Average Sentence# Per Doc
EMD	27	3,808	141 (fulltext)
ARD	146	29,687	203 (fulltext)
DD	94,654	921,396	10 (title & abstract)

As the input corpus, we used the 'Compendium of Mass Gatherings' collection on EMD that consists of 27 fulltext articles for emergency management for mass gathering [39]. This corpus was one of the main text sources used to build DO4MG [37]. On ARD, we collected 146 fulltext articles between April 2004 and April 2014 from PubMed.[5] All these articles were annotated with an indexed term, 'autism' or 'autistic disorder', and thus this assures that these are all relevant articles to autism research. For DD, we used an extensive set of biomedical articles, 94,654 articles (using only their titles and abstracts as all of their fulltext are not available), that are a 2/3 random subset of biomedical articles published from November 2012 to February 2013 in MEDLINE.[6] Table 1 summarizes the corpus used in each domain. As our evaluation metrics, we used TF presented in Formula (9).

4.2.1 Comparison with a Subsumption Method

To measure a relative performance of TaxoFinder, we compare it with a state-of-the-art subsumption method [1]. This method builds taxonomic relations based on the co-occurrences of identified concepts. The co-occurrence of two concepts x and y is identified as follows: [$P(x|y) \geq t$, $P(y|x) < t$], where t is a co-occurrence threshold. As an optimal value for t, we used 0.2 suggested by [1]. This formula is interpreted as if x appears in more than the t proportion over the documents that y appears and if y appears in less than the t proportion over the documents that x appears, x is considered a *subsumer* of y.

However, this formula allows a concept to have multiple subsumers that violates the structure of a taxonomy. To address this, [1] measures a subsumption score of a subsumer p for a given concept x to find the unique subsumer for x, denoted as $ss(p, x)$:

$$ss(p, x) = P(p|x) + \sum_{p' \in S_p} w(p', x) \cdot P(p'|x), \quad (10)$$

4. http://www.nlm.nih.gov/mesh/
5. http://www.ncbi.nlm.nih.gov/pubmed
6. http://www.nlm.nih.gov/pubs/factsheets/medline.html

where p is a potential subsumer of x, S_p is the set of subsumers of p, and $w(p', x)$ denotes a weight of the relation between p' and x, measured by $w(p', x) = 1/d(p', x)$ where $d(p', x)$ is the layer distance between p' and x. The more distant these two concepts are, the lower their weight is. If a concept has more than two subsumers, Formula (10) is applied to all the pairs of the concept and its subsumers. Then, the subsumer with the highest subsumption score is finally chosen as the a best subsumer for the concept.

4.2.2 Key Concept Extraction

As mentioned before, extracting concepts from a given corpus is the first step for taxonomy building. In this work, we used two concept extractors: a state-of-the-art key concept finder *CFinder* [10] is used for the EMD and ARD domains, and *MetaMap* [20] is used for the DD domain.

CFinder has three main steps. First, it extracts noun phrases using linguistic patterns (i.e. '(JJ)*(N)+': 'JJ' is an adjective and 'N' is a noun) based on POS tags of concept candidates for each document in the corpus. Second, it measures domain-specific weights of the candidates by combining their (1) phrase lengths, (2) word occurrence patterns, (3) statistical knowledge (i.e. frequency information in the corpus) and (4) domain-specific knowledge (i.e. domain-specific frequency information in the corpus). Third, it aggregates the weights of the candidates across all documents in the corpus, and ranks them according to their weights. Finally, the user-specified top concepts ($\mathcal{C}$) are chosen for building a taxonomy.

However, the concepts in $\mathcal{C}$ may contain particular concepts whose meanings (or senses) are possibly ambiguous. If the meanings of a certain concept are multiple (i.e. ambiguous), we can determine the most representative sense of the concept through WSD. As a WSD algorithm, we used a variant form of the SSI algorithm [29], introduced in [1]. After applying it on the corpus, we determined the most probably correct sense of each concept in its *context*, normally represented as its sentence. We also used a sentence as a unit of context. Then, we may observe that a certain concept can have different meanings in the corpus. In this case, its most frequent meaning was chosen as its correct meaning, as the taxonomy to be built will have only one meaning per concept. Also, if there are some concepts with the same meaning, we selected the highest ranked concept among them determined by CFinder as the concept label in the taxonomy, while the lower ranked concepts were filtered out from $\mathcal{C}$. This filtering approach was also used in [1]. Note that as the input of TaxoFinder and SSM, only those concepts, remained in $\mathcal{C}$ after applying the above WSD filtering approach, were used.

It is noteworthy that the SSI algorithm's results were the same as the originally extracted concepts by CFinder in both EMD and ARD domains. It means that the algorithm did not find any duplicated concept pair from the concepts to filter out. The reason comes from the following features of CFinder: it prefers to extract longer noun phrases as concepts whose constituent sub-words occur more frequently and are included in a given domain glossary. In other words, CFinder focuses on extracting longer domain-specific noun phrases than general single-words from the input corpus in general. We observed that the majority of concepts extracted by CFinder in both the EMD and ARD domains are multi-words whose length is greater than 1. We note that the SSI algorithm can be a useful method for filtering out terms whose senses are found in WordNet [1].

As described in Section 3.1, MetaMap is a tool for extracting MeSH concepts from input text through investigating their semantic relationships. More specifically, it generates a ranked list of disambiguated concepts that are estimated to be the most correctly matched MeSH terms per sentence in each document in the corpus. Thus, when we use MetaMap, we do not need to apply the WSD filtering approach discussed above. In our work, concepts recommended by MetaMap were ranked according to their occurrence frequencies over all documents in the corpus.

TABLE 2
A Summary of the CGraphs

Domain	Concept#	Edge# in CGraph	Mean Edge# Per Concept
EMD	300	9,014	30
ARD	300	5,325	18
DD	10,000	3,129,031	312

4.3 Evaluation Results and Analysis

We first examine the main features of the CGraphs and taxonomies learned by TaxoFinder on the three experimental domain. Then, we evaluate these taxonomies using gold-standard evaluation. After that we summarize the important observations drawn from the results.

4.3.1 Learned CGraphs and Taxonomies

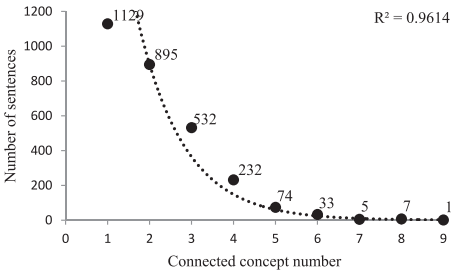
With regard to EMD, from the input corpus, we extracted the top-300 concepts by CFinder and after applying the SSI algorithm, whereas the gold-standard DO4MG ontology consists of 296 concepts.[7] Considering ARD, we also extracted the top-300 concepts using the same way as done on EMD from the input corpus, whereas the gold-standard ASDPTO ontology is comprised of 284 concepts. Regarding DD, we extracted the top-10,000 concepts using MetaMap from the large input corpus, whereas the gold-standard disease taxonomy within MeSH has 8,002 concepts.

Using the extracted concepts in each domain, we learned a taxonomy by TaxoFinder. As a size of the sliding window introduced in Section 3.2.2, we used 9 which was chosen as an optimal value in our experiments as it consistently showed the best performance in all three domains in terms of TF. To choose this, we tested TaxoFinder using odd values in [3, 11] (i.e. 3, 5,..., 11) to have the same number of preceding and following sentences for a given sentence. The maximum value 11 was chosen, since we observed that increasing values beyond 11 could not change the performance of TaxoFinder.

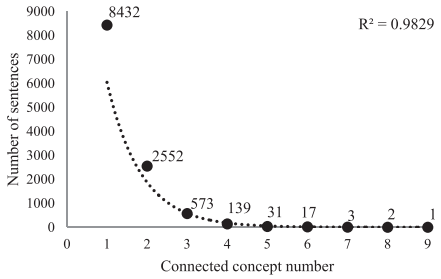
Table 2 shows some features of the constructed CGraphs.

As observed, it turned out that each concept is connected to 30-concepts on EMD, 18-concepts on ARD, and 312-concepts on DD on average, using the sliding window size 9.

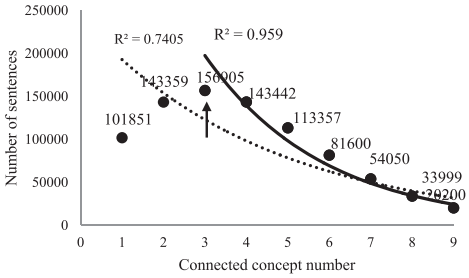
7. The version of the DO4MG ontology used in this work is an extended version of the one used in [10].



(a) EMD



(b) ARD



(c) DD

Fig. 5. The correlations between concepts appearing together and the numbers of sentences that have the occurrence of such concepts.

In the following, we present an interesting observation that the correlation between the following *two variables* highly fits into an exponential curve on each domain: (1) the number of connected concepts, that is, k-concepts (where $1 \leq k \leq 9$) appearing together in the CGraph and (2) the number of sentences that have the co-occurrence of such k-concepts.

In Figs. 5a, 5b, and 5c, we see the scatter plot diagram showing this correlation on each domain. In a diagram, *x*-axis represents the first variable, and *y*-axis (also each label attached with each plot) denotes the second variable. In a diagram, we add an exponential trend-curve (denoted in a dotted-curve) to the plotted data depicting trends in the data. Each R^2-value indicates how well the plotted data fits an exponential curve, normalized between 0 and 1. The higher a value, the better the data is fitted to the given model. As seen, the value is 0.9614 on EMD and 0.9829 on ARD. It indicates the correlations between these two variables are highly fitted to the curves overall, although the numbers of total sentences and concepts highly vary across the two domains.

On the other hand, in Fig. 5c, we see the R^2-value on DD is lower (i.e. 0.7405) than those on EMD and ARD. In particular, we see that the maximum number of sentences is 156,905 (as the arrow indicates), each containing 3-connected concepts appearing together, which is higher than the numbers of sentences each containing 1- and 2-connected concepts appearing together. We note that the input corpus for DD consists of titles and abstracts of domain articles, while that one for EMD and ARD is composed of domain fulltext articles. Thus, this observation may indicate that if the input corpus consists of fulltext articles, there may be a higher chance that the largest part of sentences have only 1-concept; while if it consists of titles and abstracts of domain articles, there may be a higher chance that the largest part of sentences have k-concepts (where $k > 1$) appearing together, where we find such k is 3. This may indicate that domain concepts are more co-located in titles and abstracts than fulltext within articles. However, in Fig. 5c, we observe that the correlation between the above two variables for k-concepts (where $3 \leq k \leq 9$) highly fits into an exponential curve as the R^2-value indicates, i.e., 0.959, calculated from the solid exponential curve. It indicates that the correlation between two variables highly fits an exponential curve for k-concepts ($k \geq 3$) in all three domains.

TABLE 3
The Five-Top Concepts in the Three Domains

Rank	EMD	ARD	DD
1	mass gathering	autism spectrum disorder	therapeutics
2	event	diagnostic and statistical manual of mental disorders (DSM-V)	population groups
3	patients	strategies for teaching based on autism research (STAR)	neoplasms
4	emergency medical services (EMS)	checklist for autism spectrum disorders in toddlers (CHAT)	evaluation studies as topic
5	patient presentation rates (PPR)	modified checklist for autism in toddlers (M-CHAT)	risk

Table 3 shows the five-top concepts identified by CFinder and MetaMap on the three domains. The top concept is 'mass gathering' on EMD, 'autism spectrum disorder' in ARD, and 'therapeutics' on DD, respectively, which is very highly relevant to each domain. As presented in Section 3.3, the top concept is used as the root in the learned taxonomy in TaxoFinder. We also note that longer concepts are extracted on ARD than EMD using CFinder. The reason is that the abbreviations used in ARD were generally comprised of longer domain terms than EMD as seen through abbreviations parenthesized in the table.

Figs. 6a and 6c show the learned taxonomies on the three domains. In each taxonomy, the larger, red-colored node represents the root concept shown in Table 3 while other concepts are denoted in gray-colored nodes. As can be seen, TaxoFinder builds a taxonomy on each domain where the root is the top ranked concept identified using a concept extractor (e.g., CFinder and MetaMap), and it has $(|\mathbf{C}| - 1)$ edges and connects all edges in $\mathbf{C}$ where $\mathbf{C}$ is the given concept set from the concept extractor.

We now compare TaxoFinder and the subsumption method introduced in Section 4.2.1. As seen in Table 4, TaxoFinder built 1 taxonomy on all three domains. On the other hand, SSM produced 46 separate sub-taxonomies on EMD, 17 separate sub-taxonomies on ARD, and three sub-taxonomies on DD. This shows that TaxoFinder is able to

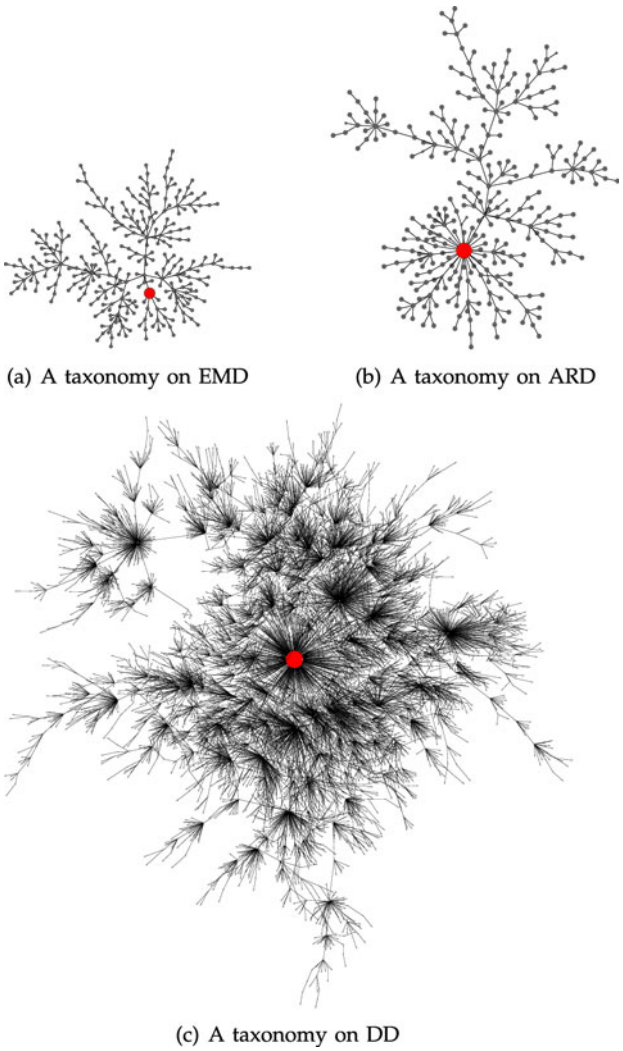


Fig. 6. The learned taxonomies. In each taxonomy, the root concept (the top-ranked concept) is denoted as the larger red-colored node.

build a taxonomy connecting all concepts together as it is derived from a graph using an MST algorithm. On the other hand, SSM could not fully consider the taxonomy connectivity. Thus, SSM may possibly have a high chance of missing important taxonomic relations between concepts, and also does not fully recognize which concept is the most general concept (i.e root).

4.3.2 Gold-Standard Evaluation

We now present the evaluation results of learned taxonomies using TF. For this, given each domain, we first built *simpler taxonomies* of the learned taxonomy T_l and gold-standard taxonomy T_g using only the identical or similar concepts commonly appearing in both taxonomies. Here, our objective is to better facilitate the comparison between these two taxonomies using such concepts.

Building these simpler taxonomies on each domain takes two steps described as follows: First, we found which concepts are in common in both T_l and T_g. For this, we selected those concepts (i.e. concept labels) in T_l that also appeared in T_g. In addition, some concepts may have the same meaning but can be labeled differently. Thus, we selected those concepts in T_l whose similar concepts also appeared in T_g. To this end, for each concept c in T_l, we measured semantic similarity between c and all concepts in T_g, and then selected only concepts highly similar to c. As the similarity measure, we used Formula (5) to measure the semantic similarity between phrases. Given two concepts, if their similarity is above 0.8, we assumed that these were highly similar to each other. Through these processes, we finally found identical or similar concepts on EMD and ARD, denoted as $\mathcal{C}$, which are in common in T_l and T_g. However, in DD, as no two MeSH terms have the same meaning, we used the exact matching not the above similarity to find the common concepts between T_l and T_g.

Second, we built simpler taxonomies of T_l and T_g using $\mathcal{C}$. Let T be a taxonomy (i.e T_l or T_g) and T^s be the T's simpler taxonomy. Given each pair $\{c_i, c_j\} \in \mathcal{C}$, if there exits a taxonomic relation between them in T, we added this pair and their taxonomic relation into T^s. By doing so, we built T^s represented using only concepts in $\mathcal{C}$ and their taxonomic relations that originally appear in T. Through the above two steps, we finally found the number of concepts and the depth of T_l and T_g for each domain as seen in Table 5.

We measure the quality of the learned taxonomies in terms of TP, TR, and TF. Table 6 shows the comparison

TABLE 5
Gold-Standard and Learned Taxonomies

Domain	Method	Concept#	Depth
EMD	Gold-Standard	48	4
	TaxoFinder		5
	SSM		3
ARD	Gold-Standard	21	4
	TaxoFinder		4
	SSM		5
DD	Gold-Standard	2,792	8
	TaxoFinder		5
	SSM		6

TABLE 4
Learned Taxonomy Comparison

Domain	Method	Taxonomy#	Depth
EMD	TaxoFinder	1	13
	SSM	46	3
ARD	TaxoFinder	1	13
	SSM	17	9
DD	TaxoFinder	1	12
	SSM	3	15

TABLE 6
Comparison between TaxoFinder and SSM

Domain	Method	TP	TR	TF
EMD	TaxoFinder	**0.58**	**0.63**	**0.61**
	SSM	0.56	0.28	0.37
ARD	TaxoFinder	**0.70**	**0.73**	**0.72**
	SSM	0.52	0.49	0.50
DD	TaxoFinder	**0.91**	**0.34**	**0.49**
	SSM	0.87	0.33	0.48

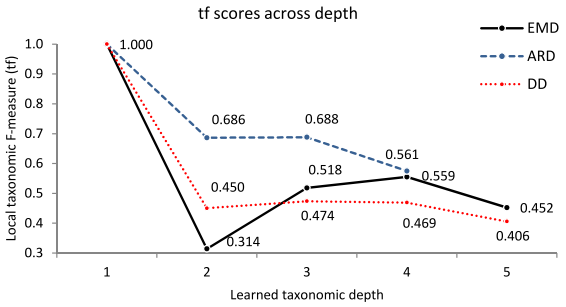


Fig. 7. The tf score changes with taxonomy depths.

between TaxoFinder and SSM on the three domains. The higher performance under each evaluation metric is denoted in bold. As seen in the table, TaxoFinder outperforms SSM on all three domains in terms of all TP, TR, and TF. In particular, in terms of TF, the quality of taxonomic relations built by TaxoFinder turns out to be approximately 61 percent, while the quality built by SSM is 37 percent on EMD; the qualities of TaxoFinder and SSM are approximately 72 and 50 percent, respectively, on ARD; and 49 and 48 percent are shown as the qualities of TaxoFinder and SSM, respectively, on DD. This roughly indicates that 61, 72, and 49 percent of the taxonomic relations, identified by TaxoFinder, are the same with the relations in the gold-standard taxonomies on EMD, ARD, and DD, respectively. On the other hand, SSM produces 37, 50, and 48 percent of taxonomic relations on EMD, ARD, and DD, respectively, in this regard. The results thus show that TaxoFinder highly outperforms SSM on all three domains: TaxoFinder shows 64.8, 44 and 2 percent improvements over SSM in terms of TF on EMD, ARD and DD, respectively.

To estimate whether TaxoFinder has significant improvements over SSM, we also performed the *paired t-test* [40]. The test demonstrated that TaxoFinder significantly outperforms SSM on all three domains at 99.9 percent ($\alpha = 0.001$) confidence (e.g., p-value $< \alpha$) with respect to TF (more exactly tf (i.e. local taxonomic F-measure) scores (see (8)). Through the above results, we show that TaxoFinder has a greater ability over SSM in building more accurate taxonomic relations.

It may also be worth examining the distribution of the tf scores of TaxoFinder across different depths of the learned taxonomy on each domain. This can help us to understand the amount of different contribution of concepts taxonomically connected to some other concepts at different depths in the taxonomy to produce the TF score in Table 6. As seen in Fig. 7, most of less accurate tf scores were made with the concepts at the 2nd-depth and the leaf concepts of the taxonomy, while the concepts at the intermediate depths generating more accurate tf scores on each domain. It may indicate that given a concept, its 'csc' (i.e. common semantic cotopy) may have more erroneous concepts at higher- or lower-depths than intermediate depths, as tf is measured based on the csc collections of the two concepts compared.

Finally, we point out the following important conclusions drawn from our evaluation. First, TaxoFinder significantly outperforms SSM. This provides evidence that TaxoFinder is an effective taxonomy learning method. Second, our evaluation results show the validity of our primary motivation of this work that utilizing both sentence distance and sentence similarity can be effectively used for building a taxonomy.[8]

4.3.3 Discussion

The aim of this work is to develop a new method for taxonomy learning by building a CGraph, which is constructed from extracted concepts from a domain corpus. The idea behind building the CGraph for taxonomy learning is that it can effectively represent a set of semantically related (in our context, co-occurred, adjacent) concepts and their relationships. In addition, implicit, meaningful relationships between some concepts can be inferred through the knowledge captured in the CGraph for taxonomy learning purposes.

The key features of TaxoFinder include combining two methods of sentence distance and sentence similarity to predict the associative strengths among concepts in a CGraph, and then applying an MST algorithm to induce a taxonomy. The distinctive features of TaxoFinder compared to existing approaches is that (1) it does not use lexico-syntactic patterns that often require additional investigation on extra knowledge sources, (2) it does not use clustering techniques that have been mostly focused on making flat clusters between concepts, and (3) it analyzes and uses intra-sentence content and inter-sentence relationships in which related concepts co-occur together.

TaxoFinder can seamlessly work together with any concept extractor able to identify a ranked list of concepts according to their relevance to the target domain such as CFinder [10]. Moreover, in the learned taxonomy, we can have more insight into how strongly or weakly associated concepts are by looking at their associative strengths measured by TaxoFinder.

In information retrieval, association knowledge between terms has been widely mined using data mining for different applications [41]. Typically, this knowledge has been synthesized mainly using co-occurrences of the terms extracted from the target corpus. For taxonomy learning, estimating associations between such terms is also essential to come up with a good taxonomy. However, most existing studies in taxonomy learning have focused on measuring these associations largely based on such co-occurrences [2]. For example, subsumption methods are generally based on the idea that a concept A subsumes a concept B if the documents (or some proportion of the corpus) that B appears are a subset of the documents that A appears [1], [19]. In another study [42], based on the co-occurrences of concepts within the corpus, a similarity measure was adopted to determine a taxonomic relation between two concepts. However, in addition to such co-occurrence notion, TaxoFinder further analyzes the impact of sentences surrounding extracted concepts using sentence similarity and sentence distance to measure their associative strengths. As shown in our evaluation, our approach can be effectively used for taxonomy learning.

5 Conclusion

This paper proposed a new taxonomy learning approach, TaxoFinder, and showed its effectiveness against a recent

8. The datasets used in the paper and more detailed snippets of our evaluation can be seen at http://yongbinkang.wix.com/main#!taxofinder/c1wcy.

subsumption method on three different domains. TaxoFinder aims to build a graph, CGraph, representing concepts extracted from a domain corpus and their associative strengths. To measure such strengths, we proposed a formula combining (1) the co-occurrence frequency of concepts within a sliding window, i.e., the set of consecutive sentences, and (2) the distance and similarity of sentences where such concepts co-occur together. From the CGraph, we used a graph analytic algorithm to induce a taxonomy aiming to maximize the overall associative strengths among concepts to find a good taxonomy. Our evaluation showed that TaxoFinder is a highly effective method for taxonomy learning, significantly outperforming the subsumption method at 99.9 percent confidence using a gold-standard evaluation method on the three domains.

In our future work, we plan to evaluate TaxoFinder with different metrics (e.g., computational complexity), and also compare TaxoFinder to hierarchical clustering methods that generate connected and deep taxonomies. We also plan to learn an optimal number of concepts as the input of TaxoFinder rather than using a fixed number of concepts. Moreover, we will attempt to apply different graph analytic methods (e.g., local connectivity of nodes) to come up with a taxonomy in addition to an MST algorithm. Further, it would be interesting to investigate incorporating Word2-Vector[9] into TaxoFinder as an alternative method to learn relationships among concepts.

ACKNOWLEDGMENTS

This research is partly supported by the Federation University 'Self-sustaining Regions Research and Innovation Initiative', an Australian Government Collaborative Research Network (CRN) grant.

REFERENCES

[1] K. Meijer, F. Frasincar, and F. Hogenboom, "A semantic approach for extracting domain taxonomies from text," *Decision Support Syst.*, vol. 62, pp. 78–93, 2014.
[2] W. Wong, W. Liu, and M. Bennamoun, "Ontology learning from text: A look back and into the future," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 20:1–20:36, Sep. 2012.
[3] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proc. 14th Conf. Comput. Linguistics*, 1992, vol. 2, pp. 539–545.
[4] P. Pantel and M. Pennacchiotti, "Espresso: Leveraging generic patterns for automatically harvesting semantic relations," in *Proc. 21st Int. Conf. Comput. Linguistics 44th Annu. Meet. Assoc. Comput. Linguistics*, 2006, pp. 113–120.
[5] X. Liu, Y. Song, S. Liu, and H. Wang, "Automatic taxonomy construction from keywords," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1433–1441.
[6] E.-A. Dietz, D. Vandic, and F. Frasincar, "TaxoLearn: A semantic approach to domain taxonomy learning," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, 2012, pp. 58–65.
[7] W. Wang, P. Mamaani Barnaghi, and A. Bargiela, "Probabilistic topic models for learning terminological ontologies," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 1028–1040, Jul. 2010.
[8] Z. Kozareva and E. Hovy, "A semi-supervised method to learn and construct taxonomies using the web," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2010, pp. 1110–1118.
[9] P. Velardi, S. Faralli, and R. Navigli, "OntoLearn Reloaded: A graph-based algorithm for taxonomy induction," *Comput. Linguistics*, vol. 39, no. 3, pp. 665–707, 2013.
[10] Y.-B. Kang, P. D. Haghighi, and F. Burstein, "CFinder: An Intelligent Key Concept Finder from Text for Ontology Development," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4494–4504, 2014.
[11] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. New York, NY, USA: McGraw-Hill, 2001.
[12] K. Dellschaft and S. Staab, "Strategies for the evaluation of ontology learning," in *Proc. Conf. Ontol. Learn. Population: Bridging Gap Between Text Knowl.*, 2008, pp. 253–272.
[13] F. M. Suchanek, G. Ifrim, and G. Weikum, "Combining linguistic and statistical analysis to extract relations from web documents," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 712–717.
[14] S. P. Ponzetto and M. Strube, "Taxonomy induction based on a collaboratively built knowledge repository," *Artif. Intell.*, vol. 175, no. 9-10, pp. 1737–1756, Jun. 2011.
[15] A. B. Rios-Alvarado, I. Lopez-Arevalo, and V. J. Sosa-Sosa, "Learning concept hierarchies from textual resources for ontologies construction," *Expert Syst. Appl.*, vol. 40, no. 15, pp. 5907–5915, Nov. 2013.
[16] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*, 4th ed. Hoboken, NJ, USA: Wiley, 2009.
[17] J. Diederich and W.-T. Balke, "The semantic growbag algorithm: Automatically deriving categorization systems," in *Proc. 11th Eur. Conf. Res. Adv. Technol. Digital Libraries*, 2007, pp. 1–13.
[18] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge Univ. Press, 2002.
[19] J. de Knijff, F. Frasincar, and F. Hogenboom, "Domain taxonomy learning from text: The subsumption method versus hierarchical clustering," *Data Knowl. Eng.*, vol. 83, pp. 54–69, 2013.
[20] A. Aronson and F.-M. Lang, "An overview of MetaMap: Historical perspective and recent advances," *J. Amer. Med. Inf. Assoc.*, vol. 17, no. 3, pp. 229–236, 2010,
[21] D. Lindberg, B. Humphreys, and A. McCray, "The unified medical language system," *Methods Inf. Med.*, vol. 32, no. 4, pp. 281–291, 1993.
[22] G. Heyer, M. Luter, U. Quasthoff, T. Wittig, and C. Wolff, "Learning relations using collocations," in *Proc. Workshop Ontol. Learning*, 2001, vol. 38.
[23] J. Seo, G.-M. Park, S.-H. Kim, and H.-G. Cho, "Characteristic analysis of social network constructed from literary fiction," in *Proc. Int. Conf. Cyberworlds*, Oct. 2013, pp. 147–150.
[24] C. Keler, M. Raubal, and K. Janowicz, "The effect of context on semantic similarity measurement," in *Proc. On the Move to Meaningful Internet Syst Workshops*, 2007, vol. 4806, pp. 1274–1284.
[25] Z. S. Harris, *Mathematical Structures of Language*. New York, NY, USA: Wiley, 1968.
[26] P. Achananuparp, X. Hu, and X. Shen, "The evaluation of sentence similarity measures," in *Proc. 10th Int. Conf. Data Warehousing Knowl. Discovery*, 2008, pp. 305–316.
[27] R. Malik, L. V. Subramaniam, and S. Kaushik, "Automatically selecting answer templates to respond to customer emails," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 1659–1664.
[28] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proc. 32nd Annu. Meeting Assoc. Comput. Linguistics*, 1994, pp. 133–138.
[29] R. Navigli and P. Velardi, "Structural semantic interconnections: A knowledge-based approach to word sense disambiguation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 7, pp. 1075–1086, Jul. 2005.
[30] H. Yang, "Constructing task-specific taxonomies for document collection browsing," in *Proc. Joint Conf. Empirical Methods Natural Language Process. Comput. Natural Language Learn.*, 2012, pp. 1278–1289.
[31] D. Cheriton and R. Tarjan, "Finding minimum spanning trees," *SIAM J. Comput.*, vol. 5, no. 4, pp. 724–742, 1976.
[32] A. Lozano-Tello, A. Gómez-Pérez, and E. Sosa, "Selection of ontologies for the semantic web," in *Proc. Int. Conf. Web Eng.*, 2003, pp. 413–416.
[33] D. Sánchez and A. Moreno, "Web-scale taxonomy learning," in *Proc. Workshop Extending Learn. Lexical Ontologies Using Machine Learn.*, 2005.
[34] P. Spyns and M.-L. Reinberger, "Lexically evaluating ontology triples generated automatically from texts," in *Proc. 2nd Eur. Semantic Web Conf.*, 2005, vol. 3532, pp. 563–577.
[35] M. Rospocher, S. Tonelli, L. Serafini, and E. Pianta, "Corpus-based terminological evaluation of ontologies," *Appl. Ontol.*, vol. 7, no. 4, pp. 429–448, Oct. 2012.

9. https://code.google.com/p/word2vec

[36] K. Liu, K. J. Mitchell, W. W. Chapman, G. K. Savova, N. Sioutos, D. L. Rubin, and R. S. Crowley, "formative evaluation of ontology learning methods for entity discovery by using existing ontologies as reference standards," *Methods Inf. Med.*, vol. 52, no. 4, pp. 308–316, 2013.
[37] P. Delir Haghighi, F. Burstein, A. Zaslavsky, and P. Arbon, "Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings," *Decision Support Syst.*, vol. 54, no. 2, pp. 1192–1204, 2013.
[38] A. McCray, P. Trevvett, and H. Frost, "Modeling the autism spectrum disorder phenotype," *Neuroinformatics*, vol. 12, no. 2, pp. 291–305, 2014.
[39] P. Arbon, "Prehospital and disaster medicine—Compendium of mass gatheirngs," *J. World Assoc. Disaster Emergency Med.*, vol. 24, 2009.
[40] D. Hull, "Using statistical testing in the evaluation of retrieval experiments," in *Proc. 16th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1993, pp. 329–338.
[41] Y.-B. Kang, S. Krishnaswamy, and A. Zaslavsky, "A retrieval strategy for case-based reasoning using similarity and association knowledge," *IEEE Trans. Cybern.*, vol. 44, no. 4, pp. 473–487, Apr. 2014.
[42] I. Novalija, D. Mladeni, and L. Bradeko, "OntoPlus: Text-driven ontology extension using ontology content, structure and co-occurrence information," *Knowl.-Based Syst.*, vol. 24, no. 8, pp. 1261–1276, 2011.

Yong-Bin Kang received the PhD degree in information technology from Monash University in 2011. He is a research associate at Monash University, Australia. His current research is focused on interdisciplinary research based on machine learning, data mining, information retrieval, and bioinformatics. He is currently working on developing a knowledge portal for autism research domain and classification models for medical articles. Prior to the PhD degree, he worked for around 10 years at various research positions in IT development companies and premier research organisations in Korea.

Pari Delir Haghighi received the PhD degree in computing in 2010 from Monash University, and then worked as a research fellow. In 2011, she received a competitive Early Career Development Fellowship (ECDF) at Monash University. Since July 2013, she has been a lecturer at Faculty of IT, Monash University. Her current research interests include mobile and context-aware computing, ontology development, rule-based reasoning, mobile healthcare, and decision support systems. She has served as a conferences program committee member, a reviewer for journal articles, and the guest editor for a special issue in the *Journal of Decision Systems*.

Frada Burstein is a professor at the Faculty of Information Technology, Monash University, Melbourne, Australia. She has been a chief investigator for a number of research projects supported by grants and scholarships from the Australian Research Council and industry, including three projects in emergency management decision support. Her current research interests include knowledge management technologies, intelligent decision support, mobile and real-time decision support, and health informatics. She is an area editor for *Decision Support Systems Journal* and a co-editor for *VINE: The Journal of Information and Knowledge Management Systems*. The most recent and substantial work was a set of two volumes of *Handbook of Decision Support Systems*, published by Springer.