Journal of the Association for Information Systems

Research Paper

ISSN: 1536-9323

Graph-based Cluster Analysis to Identify Similar Questions: A Design Science Approach

Blooma Mohan John
Faculty of Business Government and Law,
University of Canberra, Australia
blooma.john@canberra.edu.au

Alton Yeow Kuan Chua
Division of Information Studies,
Nanyang Technological University, Singapore
ALTONCHUA@ntu.edu.sg

Dion Hoe Lian Goh
Division of Information Studies,
Nanyang Technological University, Singapore
ASHLGOH@ntu.edu.sg

Nilmini Wickramasinghe
Epworth HealthCare & Faculty Health,
Deakin University, Australia
n.wickramasinghe@deakin.edu.au

Abstract:

Social question answering (SQA) services allow users to clarify their queries by asking questions and obtaining answers from other users. To enhance the responsiveness of such services, one can identify similar questions and, thereafter, return the answers available. However, identifying similar questions is difficult because of the complex language structure of user-generated questions. For this reason, we developed an approach to cluster similar questions based on a web of social relationships among the questions, the answers, the askers, and the answerers. To do so, we designed a graph-based cluster analysis using design science research guidelines. In evaluating the results, we found that the proposed graph-based cluster analysis is more promising than baseline methods.

Keywords: Cluster Analysis, Graph Theory, Design Science, Social Question Answering.

Jeffery Parsons was the accepting senior editor. This article was submitted on November 14, 2014 and went through three revisions.

1 Introduction

The advent of Web 2.0 led to the emergence of an evolving information infrastructure rich in user-generated content. The rapid growth of user-generated content has made it increasingly difficult for users to find content of interest. Arazy and Kopak (2011) highlight the sheer amount of information and its quality as major concerns today. Moreover, user-generated content also potentially leads to inaccurate, misleading, or outdated information, which researchers refer to as information waste (Amrit, Wijnhoven, & Beckers, 2015). To date, researchers have developed various analytical techniques for searching and recommending user-generated content (Adomavicius & Tuzhilin, 2005; Xu & Yin, 2015). While current search engines enjoy commercial success and demonstrate good performance, their ability to find relevant information for hard questions, such as those asking for opinions or summaries, is far from satisfactory (Harper, Moy, & Konstan, 2009). Social question answering (SQA) services satisfy these complicated user information needs. Instead of relying solely on Web search engines to search using key words, users now turn to SQA services where they find other like-minded individuals who share and meet their information needs. SQA services are dedicated platforms in which users can post their questions and respond to other users' questions (Liu et al., 2008). For example, Apple customers use Apple Store Questions & Answers to ask, answer, and rate questions related to Apple's products. WebMD exemplifies a SQA service for healthcare, and Piazza exemplifies a SQA service for collaborative learning. Yahoo! Answers, another SQA service, covers a diverse range of topics.

SQA services are a collaborative endeavor that involves group effort and open participation (Shachaf, 2010). It is interesting to look at how user-generated content in SQA services relates to not only content but also the associated users. Oh (2012) suggests that users provide answers in SQA services because of altruism or to establish their reputation as an expert in a given area. Consequently, answers contributed to SQA services range in depth depending on the answerer's technical expertise and motives. Personalized answers that other users author can be useful, especially for advice and recommendations that are difficult to answer with a general Web search. To enhance the responsiveness of such services, one can identify similar questions already found in the corpus and return the available answers. Thus, SQA services need to have an efficient mechanism to identify similar questions. However, identifying similar questions is not trivial.

SQA services are rich in multiple-sentence questions: for example, "Is it possible to download anything from YouTube? Like, music onto an iPod or onto a blank CD? If so, how?". Existing techniques to identify similar questions do not apply to or barely work in the context of such complex questions (Tamura, Takamura, & Okumura, 2005). Further, identifying similar user-generated questions collected in SQA services remains largely a challenge due to the lexical mismatch between similar questions. For example, one could also posit the question "My computer keeps displaying a blue screen and it is stuck. What should I do?" with different words as in "How to bring a frozen laptop back to life?".

In SQA services, when an asker posts a question and receives an answer from an answerer, questions and their answers form dyadic content and askers and answerers form dyadic users (Bian, Liu, Zhou, Agichtein, & Zha, 2009). Dyadic content and users result in interlinks and relationships between users and user-generated content. Hence, to overcome the lexical mismatch problem, we propose cluster analysis based on the content-user relationship.

Cluster analysis is a procedure for extricating natural configurations from content and users (Balijepally, Mangalaraj, & Iyengar, 2011). For cluster analysis, we use the relationship of a question with its answer concepts and its users to reduce the insufficiency of word similarities. To plot the relationships between questions, answers, askers, and answerers, we use graph theory. Graph theory maps the contextual information on the relationship between content and users to perform clustering analysis (Schaeffer, 2007). We use graph-based cluster analysis based on the relationships of questions with shared content (answers) and users (askers and answerers). Finally, we validate the graph-based cluster analysis using design science research (Hevner, March, Park, & Ram, 2004; Gregor & Jones, 2007). Thus, we investigate the research question

RQ: How can one identify similar questions in SQA services based on their relationship with the content shared (i.e., questions and answers) and associated users (i.e., askers and answerers)?

This paper proceeds as follows: in Section 2, we review the literature related to studies that use various techniques of cluster analysis and graph theory. In Section 3, we present the methodology we used and elaborate on the proposed graph-based cluster analysis algorithm. In Section 4, we describes our data-collection and analysis procedures. In Section 5, we present the findings, which we discuss in Section 6. Finally, in Section 7, we conclude the paper by highlighting various directions for future research.

2 Literature Review

2.1 Review of Cluster Analysis

Cluster analysis groups together similar objects into meaningful clusters based on the similarities among the objects. The algorithms used for cluster analysis are categorized into partitional and hierarchical. The partitional clustering algorithm decomposes a dataset into disjoint clusters based on a measure of dissimilarity/similarity (Steinbach, Karypis, & Kumar, 2000). For partitional algorithms, we need to specify the number of clusters, which is unpredictable when we need to find similar questions (Zhao & Karypis, 2002). The hierarchical clustering approach obtains a hierarchy of clusters through an iterative process that merges small clusters into larger ones (agglomerative algorithms) or splits large clusters into smaller ones (divisive algorithms) (Joo & Lee, 2005).

Information systems (IS) research uses cluster analysis as an analytical tool for classifying configurations of various entities that comprise the information technology artifact. Based on an analysis of 55 IS applications, Balijepally et al. (2011) reaffirm that cluster analysis is a valuable tool for IS research. However, few studies in IS research focus on clustering similar user-generated content in virtual communities such as SQA services. On the other hand, for a collection of text-based documents, existing document clustering methods include the vector space model (Salton & McGill, 1983), agglomerative cluster analysis (Walter, Bala, Kulkarni, & Pingali, 2008), the partitional k-means algorithm (Dhillon, 2001), and projection-based methods, including the least squares approximation (Kim & Park, 2008). These clustering techniques assume that words that typically appear together should be associated with similar concepts.

Although much research has studied cluster analysis of text documents (Joo & Lee, 2005), it is not directly applicable to user-generated questions from SQA services because of the nature of questions posed in these services. As users compose questions in a variety of ways, it is very likely that similar questions are worded in different ways. While one might ask a question to obtain other users' opinions, another might expect a direct answer on the same topic. Therefore, keywords alone do not provide a reliable basis for clustering user-generated questions from SQA services effectively. Moreover, similarity measures for retrieving documents based on word match work poorly when little word overlap exists (Leung, Ng, & Lee, 2008).

To overcome the disadvantages of keyword-based clustering, extant research focuses on additional criteria. One criterion is hyperlinks between documents, which builds on the hypothesis that hyperlinks connect similar documents. Beeferman and Berger (2000) use an agglomerative cluster analysis to exploit query-document relationships using click-through data. However, in their approach, the cluster analysis is content independent in the sense that it exploits only query-document links to discover similar queries and similar documents. To improve on the hyperlink concept, other studies emphasize cross-references between documents and queries in query-document clustering (Leung et al., 2008). The idea behind this type of clustering is that, if a set of queries often leads to similar documents, then those queries are similar. However, they do not consider the content. To alleviate this problem, Leung et al. (2008) introduce the notion of concept-based graphs by considering concepts extracted from Web snippets and adapt Beeferman and Berger's (2000) method to this new context. However, Leung et al. also neglect word similarity between queries. Hence, to overcome these shortcomings, we use graph theory to plot the relationship between content and users in SQA services and develop a relationship-based similarity measure to cluster similar questions.

2.2 Review of Graph Theory

Graph theory is a mathematical concept one can use to explain the properties and applications of graphs. Graphs are structures formed by a set of vertices and a set of edges that are connections between pairs of vertices. Graph clustering refers to grouping the vertices of the graphs into clusters while considering the edge structure of the graphs in such a way that there should be many edges in each cluster and relatively few between the clusters (Schaeffer, 2007).

Mapping Web-based social interactions onto a graph represents a classical example of applying graph theory to complex networks (Boccaletti, Latora, Moreno, Chavez, & Hwang, 2006). Web-based social interactions are a multipartite network representation. The vertices of the graph are represented by the interacting agents (humans, users of the web portals) and the subjects of their interactions (music, movies, books, postings) in a social network. One then analyzes their mutual connections in detail. For example, Lambiotte and Ausloos (2005, 2006) use social connections related to music to detect communities related to music genres.

Bipartite networks comprise two kinds of vertices. Some different types of vertices used in bipartite networks are query-document (Rege, Dong, & Fotouhi, 2006), query-URL (Li, Yuan, & Jing, 2007), user-movie (Grujic, Mitrovic, & Tadic, 2009), and question-answer (Bian et al., 2009). Researchers have used bipartite networks for various cluster analysis applications, such as mining text (Leung et al., 2008; Li et al., 2007), mapping ontologies (Chen & Fonseca, 2003), identifying user communities (Grujic et al., 2009), extracting verb synonyms (Takeuchi, 2008), and clustering reliable users and content in social media (Bian et al., 2009). Beeferman and Berger (2000) use bipartite graphs for clustering queries using hyperlinks. Dhillon (2001) and Rege et al. (2006) use bipartite graphs for co-clustering documents. Wen, Nie, and Zhang (2002) and Li et al. (2007) use bipartite graphs for query clustering, and Leung et al. (2008) use a bipartite graph for concept-based query clustering. However, none of these studies considers query-content similarity or user similarity. On the other hand, Bian et al. (2009) use a mutually coupled bipartite network to identify high-quality content and users. They consider interactions in SQA services as composite bipartite graphs and use the mutual reinforcement between the connected entities in each bipartite graph to compute their respective quality and reputation scores.

Tripartite networks comprise three kinds of vertices. Two types of vertices used in tripartite networks are user-resource-tags (Lambiotte & Ausloos, 2006) and visual feature-Web image-related text (Rege, Dong, & Hua, 2008). Researchers have used tripartite networks for various applications, such as collaborative tagging (Lambiotte & Ausloos, 2006), Web clustering (Lu, Chen, & Park, 2009), and Web image clustering (Rege et al., 2008). Networks formed using tripartite graphs are superior to bipartite networks because they consider the possibility of correlations among three kinds of vertices. Among studies related to cluster analysis, Lambiotte and Ausloos (2006) use users, resources, and tags as vertices in a tripartite network for collaborative tagging. Lu et al. (2009) use similar vertices to investigate how to enhance Web clustering by leveraging the tripartite network of social tagging systems. Rege et al. (2008) propose a tripartite network of visual and textual features of images for efficient Web image clustering. They address the semantic gap between visual features and high-level semantic concepts to overcome the shortcomings of Web image clustering.

Thus, based on our review, we see that a need to use content and relationship between questions, answers, and users to identify similar questions exists. Moreover, the literature review forms the foundation for the graph-based cluster analysis for clustering similar questions. The design of graph-based cluster analysis is this study’s novel contribution to the research domain, and we present it in Section 3 using design science research components.

3 Methodology

Design science is an important and legitimate IS research paradigm (Gregor & Hevner, 2013). In IS, design science research involves constructing a wide range of socio-technical artifacts, such as new software, processes, algorithms, or systems intended to improve or solve an identified problem (Myers & Venable, 2014). While we present the quadripartite graph-based cluster analysis, the nature of design science research provides a foundation for more systematically specifying its design knowledge.

Hevner et al. (2004) present seven guidelines for understanding, executing, and evaluating design science research. Various studies (Arnott & Pervan, 2012; Xu, Wang, Li, & Chau, 2007) use these guidelines and we explain how we used the guidelines for this study in Table 1. Figure 1 illustrates the steps that result in identifying similar questions in SQA services.

As Figure 1 shows, first, we collected questions, answers, askers, and answerers from the SQA corpora. Thus, for this study, we used data from four popular categories of Yahoo! Answers (i.e., “arts and humanities”, “business and finance”, “computers and Internet”, and “science and mathematics”). We restricted the questions to the resolved section because one can expect a question in this section to contain the “best” answer (i.e., the answer that the asker preferred). Second, we extracted and pre-processed the required features. The final dataset for clustering questions after pre-processing contained a total of 5,733 questions from the four categories. The average length of processed questions and answers was 4.5 words and 29.6 words, respectively. Third, we executed the method artifact. The method artifact includes two steps: 1) graph network formation and 2) cluster analysis of similar questions.

Table 1. Design Science Research Guidelines

Guidelines	Description (Hevner et al. 2004)	Used in this study
Design as an artifact	Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.	The design artifact is an instantiation of a method to conduct graph-based cluster analysis.
Problem relevance	Design science research focuses on developing technology-based solutions to important and relevant business problems.	We focused on developing graph-based cluster analysis to solve the complexity of searching through user-generated content in social media.
Design evaluation	One must rigorously demonstrate the utility, quality, and efficacy of a design artifact via a well-executed evaluation method.	After testing the nine different combinations of algorithms for three different similarities, we evaluated the performances of different clustering algorithms using precision, recall, and the f-measure.
Research contributions	Effective design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.	We applied graph theory to form a question-answer-asker-answerer quadripartite network and to identify similar questions.
Research rigor	Design science research relies on applying rigorous methods in both constructing and evaluating the design artifact.	We constructed a quadripartite network, designed the variations of the clustering algorithm, and evaluated each algorithm.
Design as a search process	The search for an effective artifact requires using the means available to reach the desired ends while satisfying laws in the problem environment.	The quadripartite clustering algorithm used an established agglomerative clustering. The algorithm extended agglomerative clustering by using graph theory.
Communication of research	One must present design science research effectively both to technology-oriented and management-oriented audiences.	We published at various IS conferences and discussed the algorithm and its applicability in education and healthcare social media.

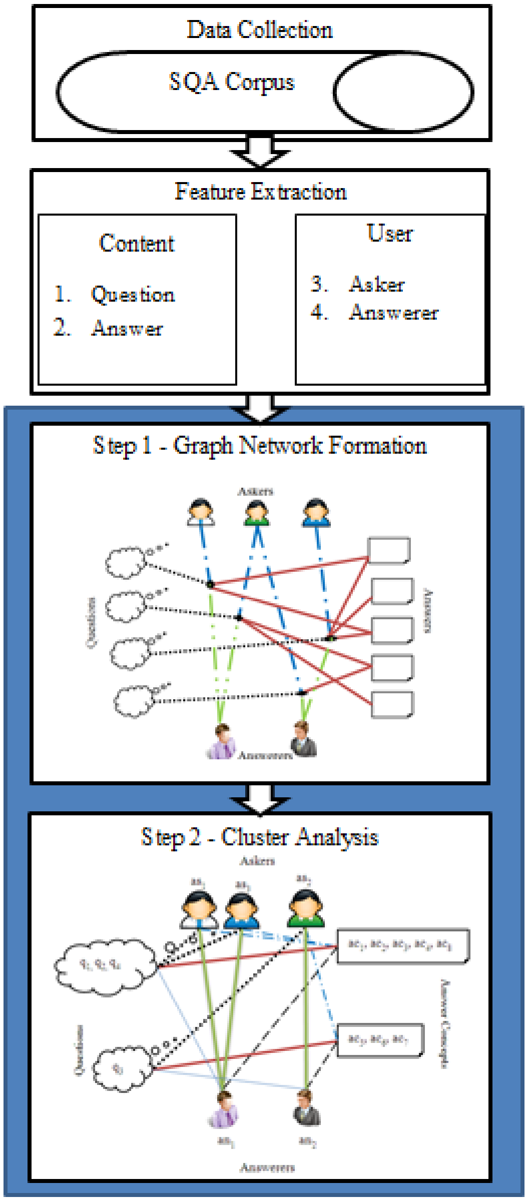

Figure 1. Design Artifact for Graph-based Cluster Analysis

3.1 Step 1: Graph Network Formation

We formed a quadripartite network with concepts extracted from the best answer, asker profile, and answerer profile related to a question. We call this network a question-answer-asker-answerer quadripartite network. The quadripartite structure of a question-answer-asker-answerer network differs fundamentally from the bipartite structure of well-studied hyperlink or concept-query graphs (Beeferman & Berger, 2000; Leung et al., 2008). The basic assumptions we used for this approach are as follows: similar questions lead to similar answers, and similar askers will have similar information needs and, hence, will pose similar questions. Similar askers will prefer similar answers. Similar answerers will answer similar questions. We based these assumptions on the coupled mutual reinforcement principle that Bian et al. (2009) propose.

The vertices in a question-answer-asker-answerer network have an in-depth relationship. The users can be either askers, or answerers, or both (Bian et al., 2009). Given the potentially different roles users play, we used quadripartite network to uncover similar questions based on askers who asked similar questions and answerers who answered similar questions. Hence, we considered both askers and answerers as vertices in the quadripartite graph to identify similar questions based on question-answer-asker-answerer relationships.

SQA services focus on four entities (questions (Q), answers (A), askers (As), and answerers (An)) and their relationships. We call the relationship between these four entities a quadripartite network, and we represent it in this study as a question-answer-asker-answerer network. Figure 2 provides an example of the quadripartite network of an SQA service.

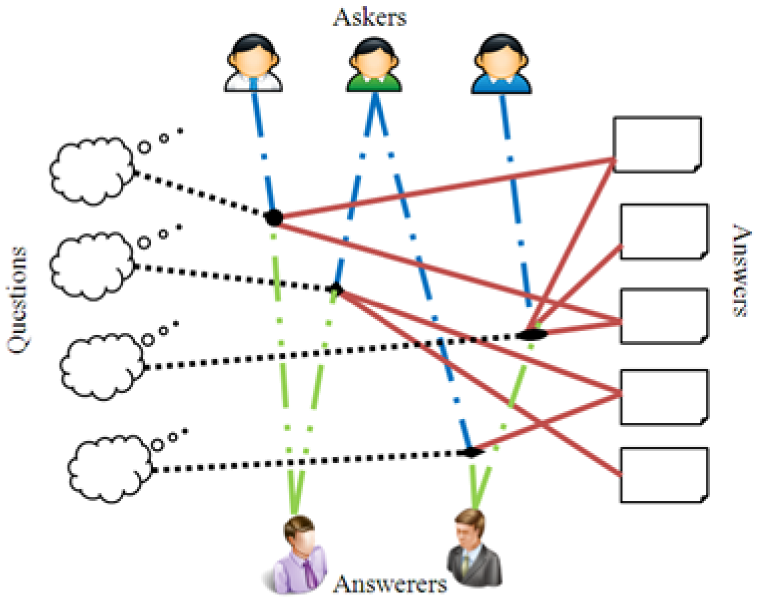

Figure 2. An Example of a Quadripartite Network for SQA

To construct a quadripartite graph, we represented answers in the network as answer concepts in the graph after extracting keywords or phrases from the best answer to represent its important semantic concepts (Leung et al., 2008). We used the intricate network of relationships among questions, answer concepts, askers, and answers to cluster similar questions. In the quadripartite graph, the first side of the vertices corresponds to questions, the second side corresponds to answer concepts, the third side corresponds to askers, and the fourth side corresponds to answerers. If an asker who asks a question receives a best answer from an answerer, they form six different types of links. We classified links formed between the vertices according to the entity as we detail below:

- Questions
  1. A question is linked with its corresponding answer concepts extracted from its best answer. For example, question *q_1* is linked to answer concept *$ac_{1.}$*
  2. A question is linked to its corresponding asker. For example, question *q_1* is linked to asker *as_1*.
  3. A question is linked to its corresponding answerer. For example, question *q_1* is linked to answerer *an_1*.
- Answer concepts
  4. An answer concept is linked to its respective answerer. For example, answer concept *ac_1* is linked to answerer *an_1*.
  5. An answer concept is linked to its respective question's asker. For example, answer concept *ac_1* is linked to asker *as_1.*
- Askers and answerers
  6. An asker is linked to their respective answerer. For example, asker *as_1* is linked to answerer *an_1*.

We then converted the quadripartite network into a quadripartite graph using Algorithm 1 (see below) based on the six different types of links. Figure 3 illustrates the resulting quadripartite graph.

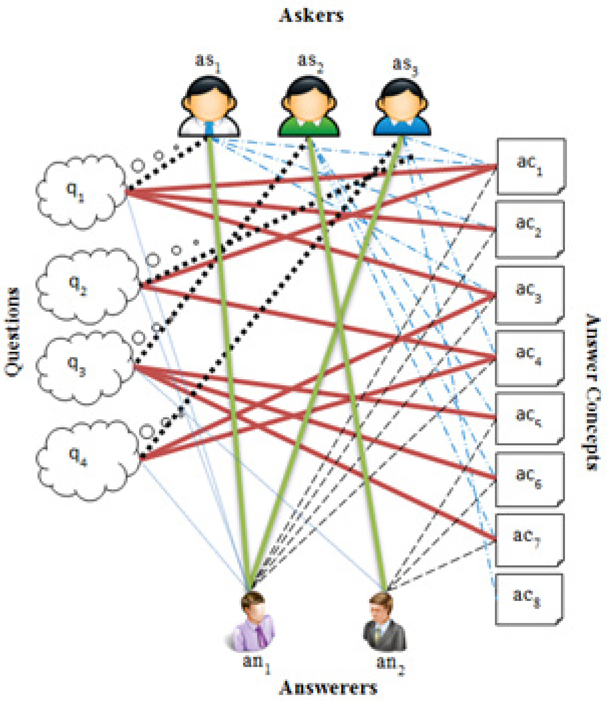

Figure 3. An Example of the Quadripartite Graph for SQA

3.1.1 Algorithm 1: Quadripartite Graph (QG) Construction

Input: Question (*Q*) and its related answer concepts (*AC*) and its asker (*AS*) and answerer (*AN*) relationships are collectively termed *QRelation*.

Output: A question-answer-asker-answerer quadripartite graph *(QG)*

1) Obtain the set of unique questions $Q = \{q_1, q_2, q_3, \ldots\}$ from *QRelation*.
2) Obtain the set of unique answer concepts $AC = \{ac_1, ac_2, ac_3, \ldots\}$ from answers $A = \{a_1, a_2, a_3 \ldots\}$ in *QRelation*.
3) Obtain the set of unique askers $AS = \{as_1, as_2, as_3, \ldots\}$ from *QRelation*.
4) Obtain the set of unique answerers $AN = \{an_1, an_2, an_3, \ldots\}$ from *QRelation*.
5) The total number of vertices in $QG = Q \cup AC \cup AS \cup AN$, where *Q, AC, AS,* and *AN* are the four sides in *QG*.
6) If answer a_i, answered by the answerer an_l, is marked as the best answer by the asker as_k for the question $q_i \in Q$, the following edges will be created:
   i. an edge $e_1 = (q_i, ac_j)$ in QG for all answer concepts ac_j in answer a_i to question q_i
   ii. an edge $e_2 = (q_i, as_k)$ in *QG* where as_k is the asker of question q_i
   iii. an edge $e_3 = (q_i, an_l)$ in *QG* where an_l is the answerer of answer a_i
   iv. an edge $e_4 = (ac_j, as_k)$ in *QG* for all answer concepts ac_j in answer a_i to question q_i asked by asker as_k

v. an edge $e_5 = (ac_j, an_l)$ in *QG*, for all answer concepts ac_j in answer a_i answered by answerer an_l, *and*

vi. an edge $e_6 = (as_k, an_l)$ in *QG*.

As an example, we present four sample questions (see Figure 4). The example has three askers and two answerers. From the figure, one can see that one asker asked two questions. One answerer answered three questions. We extracted eight answer concepts from the answers. For the first question "What is LIC?", the important concepts its answer uses are "life", "return", and "insurance". Answers to the second question also include the concept "life". Thus, based on the graph, we identified the most similar questions based on the common answer concepts used or the users involved.

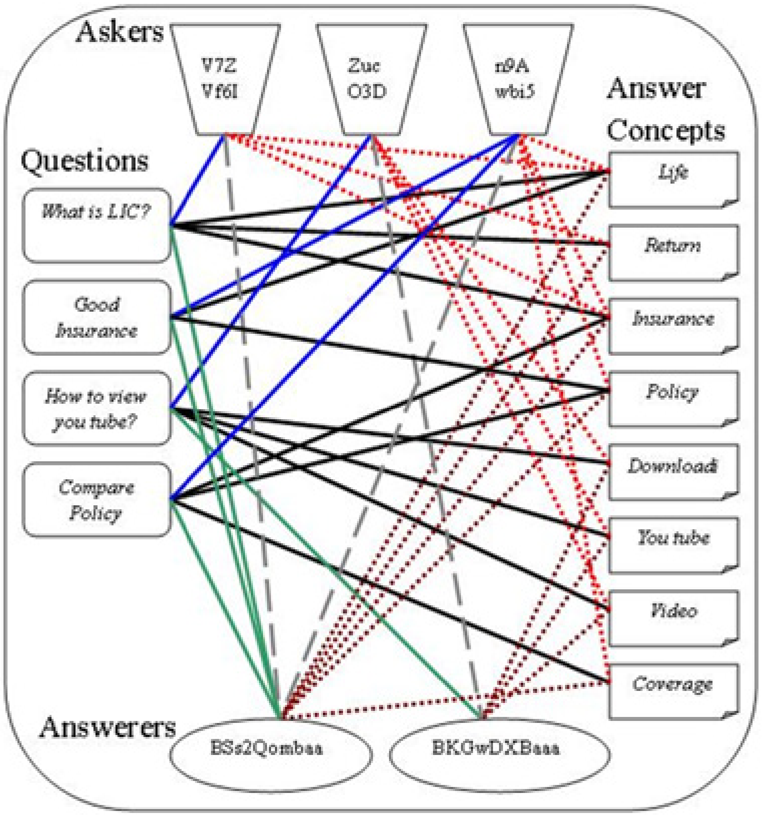

Figure 4. A Quadripartite Graph Constructed from Four Sample Questions

3.2 Step 2: Cluster Analysis

After constructing the quadripartite graph, we used cluster analysis to identify similar questions. The cluster analysis depends on the clustering variables, the similarity measure, and the clustering algorithm used in the analysis. In particular, we used quadripartite graph-based cluster analysis to identify similar questions. Figure 5 provides the outcome of the quadripartite graph-based cluster analysis. The cluster analysis clusters the three questions on insurance policies based on similar askers, answerers, and answer concepts. In Section 4, we describe in detail the algorithm we used.

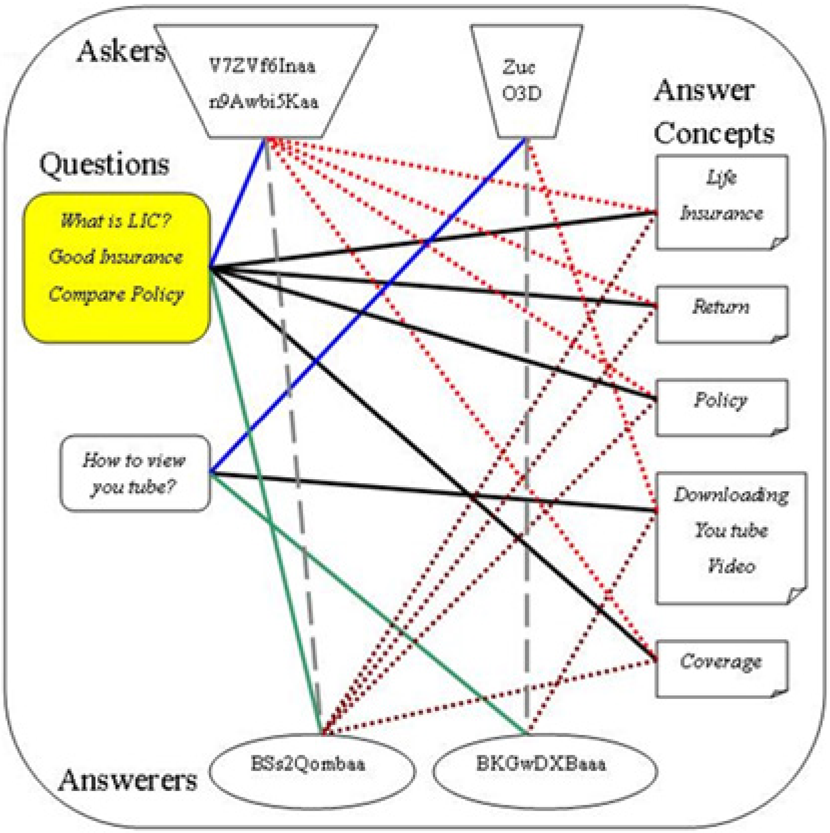

Figure 5. The Result of Quadripartite Graph-based Cluster Analysis

4 Cluster Analysis

Cluster analysis involves grouping together similar vertices into groups based on similarities among the vertices in the quadripartite network. As Balijepally, Mangalaraj, and Iyengar (2011) suggest, one should follow five steps in applying cluster analysis: 1) clustering variables, 2) the similarity measure, 3) clustering algorithms, 4) determining the number of clusters, and 5) validating clusters. We followed these five steps to design the cluster analysis.

4.1 Clustering Variables

Selecting variables is an important step because the variables define the features that structure the clustering process. One needs to meet two conditions when selecting variables (Balijepally et al., 2011). First, one should draw the variables selected for describing the groups from past research or theory. Second, the variables selected must be consistent with one's study's objectives. In compliance with the two conditions, the variables we selected for this study were questions, answers, askers, and answerers. We considered four variables as vertices in a quadripartite graph and the network of relationships between them. Moreover, Bian et al. (2009) use these four variables to form a mutually coupled bipartite network to identify quality content and reputed users. In our study, we focus on identifying similar questions based on their relationship with shared content and users. Thus, questions and the related best answer represent shared content and askers and answerers represent related users.

4.2 The Similarity Measure

Selecting an empirical measure of similarity between the entities is an important research decision. As different similarity measures may produce different clusters, researchers often recommend using several similarity measures and comparing the cluster with theoretical or known patterns (Hair, Black, Babin, Anderson, & Tatham, 2006). We tested four different types of similarity measures (see Sections 4.2.1 to 4.2.4). We adopted the first and second similarity measures from extant literature. We designed the third and fourth similarity measures to evaluate the graph-based cluster analysis technique.

4.2.1 Question Content Similarity ($sim_{QC}(q_i,q_j)$)

The first similarity measure considers only question content. We calculated the similarity matrix using cosine similarity for words (Salton & McGill, 1983). We used cosine similarity for two reasons: their popularity and their suitability for high dimensional data (Wu et al., 2008; Lu et al., 2009). To calculate cosine similarity, we defined a set of questions as shown in Equation 1. We converted a single question, $q_{j,}$ to a term and weight vector as shown in Equation 2. In Equation 2, q_i is an index term of q_j and w_{iqj} represents the weight of the i^{th} term in question q_j. The question frequency qf_i is defined as the number of questions in a collection of n questions that contains the term q_i. A high term frequency indicates that a term is highly related to a question and, thus, is more important in the clustering process. A high question frequency, on the other hand, indicates that a term is too general to be useful as a descriptor and will not convey useful information for question clustering. Next, we computed the inverse question frequency iqf_i as shown in Equation 3 in which n represents the total number of questions in the question collection. We then computed the weight of the term w_iq_j based on Equation 4. The term frequency tf_iQ_j is the frequency of word w_i in question Q_j and is used for computing the weight of the term. We finally computed the cosine similarity as shown in Equation 5.

$$Q=\{q_1, q_2, q_3 \ldots q_i, q_j, \ldots q_n\} \tag{1}$$

$$Q_j = \{<q_1,w_{1Q}>; <q_2,w_{2Q}>; \ldots\ldots <q_i,w_{iQ}>\} \tag{2}$$

$$iqf_{\,i} = \log(n/qf_i) \tag{3}$$

$$w_iQ_j = tf_iQ_j * iqf_i \tag{4}$$

$$\mathrm{sim}_{QC}(Q_i, Q_j) = \frac{\sum_{i=1}^{k} \mathrm{cw}_{iQi} * \mathrm{cw}_{iQj}}{\sqrt{\sum_{i=1}^{k} \mathrm{w}_{iQi}{}^{2}} * \sqrt{\sum_{i=1}^{k} \mathrm{w}_{iQj}{}^{2}}} \tag{5}$$

In Equation 5, cw_{iQi} refers to the weight of the i^{th} common term of C_{ij} in a question.

4.2.2 Question and Answer Relationship Similarity ($sim_{qar}(q_i,q_j)$)

The second similarity measure considers the question and answer relationship. We adapted the question and answer relationship similarity measure from Leung et al. (2008) for finding similar questions on the quadripartite graph QG. In other words, by adapting this formula to a bipartite graph of answer concepts and questions, we could define $sim_{qar}(q_i, q_j)$ as the similarity between two questions represented by vertices q_i and q_j. We calculated the value $sim_{qar}(q_i,q_j)$ as the number of links to common answer concept vertices divided by the total number of unique links from q_i and q_j (Equation 6). Intuitively, the similarity measure formalizes the idea that q_i and q_j are similar if their respective neighboring vertices largely overlap and vice versa.

$$sim_{qar}(q_i, q_j) = \left\{ \begin{matrix} \dfrac{|L(q_i, q_j)|}{|L(q_i)UL(q_j)|}, \\ 0 \end{matrix} \right\} \tag{6}$$

In Equation 6:

- $L(q_i, q_j)$ is a set of links connecting q_i and q_j to the same vertices
- $L(q_i)$ and $L(q_j)$ are all the links connecting to q_i and q_j, respectively, and
- $|L(.)|$ is the cardinality of $L(.)$; $(.)$ stands for q_i or q_j or q_i and q_j.

4.2.3 Question, Answer, Asker, Answerer Relationship Similarity ($sim_{qr}(q_i,q_j)$)

The third similarity measure considers the relationship between a question, answer, asker, and answerer. We extended the question and answer relationship similarity to the quadripartite graph with four vertices by modifying the similarity measure with respect to the question's relationship with the answer, asker and answerer vertices. For example, to calculate question similarity based on its relationship with other vertices, we considered the overlap of answer concepts, askers, and answerers. The similarity measure, based on a relationship denoted as $sim_{qr}(q_i,q_j)$, is shown in Equation 7. Equation 8 gives the similarity measure of answer concepts, $sim_{acr}(ac_i,ac_j)$. We do not provide the asker, $sim_{asr}(as_i,as_j)$, and answerer, $sim_{anr}(an_i,an_j)$, similarity measures due to space limitations. For both askers and answerers, we used the same relationship parameters α, β, and γ for question, answer, and asker/answerer, respectively.

$$sim_{q_r}(q_i,q_j) = \begin{cases} \alpha\left\{\frac{|L_{ac}(q_i,q_j)|}{|L_{ac}(q_i)UL_{ac}(q_j)|}\right\} + \beta\left\{\frac{|L_{as}(q_i,q_j)|}{|L_{as}(q_i)UL_{as}(q_j)|}\right\} + \gamma\left\{\frac{|L_{an}(q_i,q_j)|}{|L_{an}(q_i)UL_{an}(q_j)|}\right\} & \textit{if } |L_{ac}(q_i)UL_{ac}(q_j)| > 0, \\ & |L_{as}(q_i)UL_{as}(q_j)| > 0, \\ & |L_{an}(q_i)UL_{an}(q_j)| > 0 \\ 0 & \textit{Otherwise} \end{cases} \quad (7)$$

In Equation 7:

- *L_{ac}(q_i, q_j)* is the set of links connecting questions *q_i* and *q_j* to the same vertices of answer concepts
- *L_{ac}(q_i)* and *L_{ac}(q_j)* are all the links connecting to *q_i* and *q_j*, respectively, from the vertices of answer concepts
- *L_{as}(q_i, q_j)* is the set of links connecting *q_i* and *q_j* to the same vertices of askers
- *L_{as}(q_i)* and *L_{as}(q_j)* are all the links connecting to *q_i* and *q_j*, respectively, from the vertices of askers
- *L_{an}(q_i, q_j)* is the set of links connecting *q_i* and *q_j* to the same vertices of answerers
- *L_{an}(q_i)* and *L_{an}(q_j)* are all the links connecting to *q_i* and *q_j*, respectively, from the vertices of answerers,
- |*L(.)*| is the cardinality of *L(.)*, and
- *α*, *β*, and *γ* are relationship parameters.

$$sim_{ac_r}(ac_i,ac_j) = \begin{cases} \alpha\left\{\frac{|L_q(ac_i,ac_j)|}{|L_q(ac_i)UL_q(ac_j)|}\right\} + \beta\left\{\frac{|L_{as}(ac_i,ac_j)|}{|L_{as}(ac_i)UL_{as}(ac_j)|}\right\} + \gamma\left\{\frac{|L_{an}(ac_i,ac_j)|}{|L_{an}(ac_i)UL_{an}(ac_j)|}\right\} & \textit{if } |L_q(ac_i)UL_q(ac_j)| > 0, \\ & |L_{as}(ac_i)UL_{as}(ac_j)| > 0, \\ & |L_{an}(ac_i)UL_{an}(ac_j)| > 0 \\ 0 & \textit{Otherwise} \end{cases} \quad (8)$$

In Equation 8:

- *L_q(ac_i, ac_j)* is the set of links connecting answer concepts *ac_i* and *ac_j* to the same vertices of questions
- *L_q(ac_i)* and *L_q(ac_j)* are all the links connecting to *ac_i* and *ac_j*, respectively, from the vertices of questions
- *L_{as}(ac_i, ac_j)* is the set of links connecting *ac_i* and *ac_j* to the same vertices of askers
- *L_{as}(ac_i)* and *L_{as}(ac_j)* are all the links connecting to *ac_i* and *ac_j*, respectively, from the vertices of askers
- *L_{an}(ac_i, ac_j)* is the set of links connecting *ac_i* and *ac_j* to the same vertices of answerers
- *L_{an}(ac_i)* and *L_{an}(ac_j)* are all the links connecting to *ac_i* and *ac_j,* respectively, from the vertices of answerers
- |*L(.)*| is the cardinality of *L(.)*, and
- *α, β,* and *γ* are relationship parameters*.*

4.2.4 Content and Relationship Similarity ($sim_{q_{c+r}}(q_i,q_j)$)

The fourth similarity is a combined similarity measure that considers the similarity of content and the question, answer, asker, and answerer relationship. We further enhanced the relationship similarity we discuss above using a combined similarity measure by including question word similarity. We required this enhancement because similarity measures based on a question's content and the question-answer-asker-answerer relationship represent two different viewpoints. First, content-based similarity measures tend to cluster questions with the same or similar terms, but one could use similar terms to represent different requirements because of the ambiguity of words (Wen et al., 2002). Second, similarity measures based on the question-answer-asker-answerer relationship tend to cluster questions related to the same or similar topics. However, an asker or answerer might have more than one topic of interest. One might use answer concepts in different contexts. Thus, questions could lead to answers containing the same answer concepts or the same askers and answerers (Bian et al., 2009). Since each of the above criteria partially capture a user's information needs, we extended the quadripartite algorithm based on the question-answer-asker-answerer relationship to consider the question's words. Hence, to implement quadripartite clustering using a combined similarity, we created the new combined similarity measure:

$$sim_{qc+r}(q_i, q_j) = \delta * sim_{content}(q_i, q_j) + (1 - \delta) * sim_{qr}(q_i, q_j), \quad (9)$$

where δ is a constant called the content parameter.

We obtained the question similarity of content $sim_{content}$ from the commonly used content similarity measure (Wen et al., 2002) given as:

$$sim_{\text{content}}(q_i, q_j) = \frac{N(q_i, q_j)}{Max(N(q_i), N(q_j))}, \quad (10)$$

where

- N(.) is the number of keywords in a question and
- $N(q_i, q_j)$ is the number of common keywords in two questions.

4.3 Clustering Algorithm

The clustering algorithm has a significant impact on the cluster analysis. In this study, we used the hierarchical clustering algorithm (HCA) for clustering the quadripartite graph for two reasons. First, HCA is more flexible because it supplies an arbitrary function that defines what constitutes a good pair to cluster together (Walter et al., 2008), which is especially convenient for data combining different types of properties and in higher dimensions, such as a quadripartite graph structure. Second, in a partitional algorithm, one needs to specify the number of clusters a priori, which is unpredictable in the case of the SQA corpora (Wu et al., 2008). Similarly, the more commonly used density-based spatial clustering of applications with noise for large datasets requires one to provide more precise termination conditions. Further, HCA is the most popularly used clustering algorithm in IS research (Balijepally et al., 2011).

From the various HCA algorithms available, we used the complete link algorithm for this study because previous studies recommend it as one of the most effective in terms of optimal cluster evaluation (Tombros, Villa, & Van Rijsbergen, 2002). The complete link algorithm also demonstrates a lower computational complexity than other HCA algorithms. In the complete link algorithm, the similarity of two clusters is the similarity of their most dissimilar vertices. The complete link algorithm produces tightly bound or compact clusters. Based on the number of vertices involved, we compared four variations of HCA listed as agglomerative, bipartite, quadripartite, and improved quadripartite.

4.3.1 Agglomerative Cluster Analysis

The agglomerative algorithm accomplishes hierarchical clustering by assigning objects to their own cluster and then repeatedly merging pairs of clusters until it forms the whole dendogram. For this study, we used agglomerative cluster analysis as the baseline. We calculated the similarity matrix using cosine similarity for words as given in Equation 5. Agglomerative cluster analysis using question phrases is a more precise representation of meaning than question words. We represent agglomerative cluster analysis as AC.

4.3.2 Bipartite Graph-based Cluster Analysis

We adapted the algorithm for bipartite graph-based cluster analysis from Leung et al. (2008). We used concepts extracted from the best answers to form a question-answer concept bipartite graph. We obtained answer concepts from the best answers by extracting the noun phrases. We adopted the similarity measure proposed by Leung et al. (2008), sim_{QAR}, given in Equation 6. We represent the bipartite graph-based cluster analysis as BR.

4.3.3 Quadripartite Graph-based Cluster Analysis

We used quadripartite graph-based cluster analysis algorithm of Blooma and Kurian (2012) (Algorithm 2). We tested quadripartite graph-based cluster analysis with both relationship similarity, $sim_{qr}(q_i,q_j)$, as well as combined similarity, $sim_{q_{c+r}}(q_i, q_j)$. We represent the quadripartite graph-based cluster analysis with the relationship similarity as QR and quadripartite graph-based cluster analysis with the combined similarity as QRC.

Algorithm 2: quadripartite graph-based cluster analysis:

Input: A question-answer-asker-answerer quadripartite graph (*QG*)

Output: A clustered question-answer-asker-answerer quadripartite graph (QG_c)

1. Obtain the similarity scores for all possible pairs of questions in *QG* using the noise-tolerant similarity measure $sim_{qr}(q_i,q_j)$.
2. Merge the pair of questions (q_i, q_j) that has the highest similarity score.
3. Obtain the similarity scores for all possible pairs of answer concepts in *QG* using the noise-tolerant similarity measure $sim_{acr}(ac_i,ac_j)$).
4. Merge the pair of answer concepts (ac_i, ac_j) that has the highest similarity score.
5. Obtain the similarity scores for all possible pairs of askers in *QG* using the noise-tolerant similarity measure $sim_{asr}(as_i,as_j)$.
6. Merge the pair of askers (as_i, as_j) that has the highest similarity score.
7. Obtain the similarity scores for all possible pairs of answerers in *QG* using the noise-tolerant similarity measure $sim_{anr}(an_i,an_j)$.
8. Merge the pair of answerers (an_i, an_j) that has the highest similarity score.
9. Unless termination condition is reached, repeat steps 1-8.

4.3.4 Improved Quadripartite Graph-based Cluster Analysis

We designed improved quadripartite graph-based cluster analysis to reduce the time taken to complete the clustering process as compared to quadripartite graph-based cluster analysis (Blooma & Kurian, 2012; Blooma, Chua, & Goh, 2011). In the improved quadripartite graph-based clustering, after calculating the question similarity, we grouped the questions and answer concepts, askers, and answerers. The algorithm, Algorithm 3, is a significant contribution of this study. We tested the algorithm with the relationship similarity, $sim_{qr}(q_i,q_j)$. We represent the improved quadripartite graph-based cluster analysis as *IQR*.

Algorithm 3: improved quadripartite graph-based cluster analysis:

Input: A question-answer-asker-answerer quadripartite graph (*QG*)

Output: A clustered question-answer-asker-answerer quadripartite graph (QG_c)

1. Obtain the similarity scores for all possible pairs of questions in *QG* using the noise-tolerant similarity measure given in Equation 7.
2. Merge the pair of questions (q_i, q_j) that has the highest similarity score.

3. Merge the respective pair of answer concepts (ac_i, ac_j) of questions (q_i, q_j) that has the highest similarity score.

4. Merge the pair of askers (as_i, as_j) of questions (q_i, q_j) that has the highest similarity score.

5. Merge the pair of answerers (an_i, an_j) of questions (q_i, q_j) that has the highest similarity score.

6. Unless termination condition is reached, repeat steps 1-5.

4.4 Determining the Number of Clusters

As no standard procedures exist to help select the number of clusters, one needs to select the number of clusters that best represent their data's underlying structure. Moreover, an increase in the heterogeneity of clusters accompanies a decrease in the number of clusters. As Hair et al. (2006) suggest, we used three cut-offs (threshold values of 0.50, 0.70, and 0.90) in computing cluster solution sizes and comparing the performance at various stages. These thresholds represent a similarity of 50 percent, 70 percent, and 90 percent. The limit of three thresholds reduces the computational complexity (Chen & Fonseca, 2003). We used the higher threshold values 0.50, 0.75, and 0.90 to obtain question clusters with higher similarity. We considered the three threshold values of equal intervals to observe the performance in three distinct ranges.

4.5 Validation of Clusters

As Punj and Stewart (1983) highlight, one needs to ensure they validate clusters' meaningfulness and utility. One establishes reliability, a prerequisite for validity, by checking the stability of cluster solutions by using multiple algorithms (Hair et al., 2006). Thus, we used nine combinations of cluster analysis to establish validity and reliability. Quadripartite graph-based cluster analysis with the relationship similarity (QR) uses three different combinations of α, β, and γ values to determine which combination of relationship parameters generates the best performance. Hence, we set the values of QR_1 at {0.80, 0.10, 0.10}, QR_2 at {0.34, 0.33, 0.33}, and QR_3 at {0.20, 0.40, 0.40}. By giving various combinations of α, β, and γ values, we analyzed various relationships between the questions, answers, askers, and answerers. For example, consider QR_1. QR_1 signifies that the question similarity measure combines 80 percent of the answer concept and 10 percent of the asker and 10 percent of the answerer relationships to obtain the total similarity measure. We used the best results from the three combinations of weights to further proceed with combined similarity and improved quadripartite graph-based cluster analysis.

In our quadripartite graph-based cluster analysis with the combined similarity (QRC), we evaluated three different combinations of the content parameter δ to determine which combination generated the best performance. QRC_1, QRC_2, and QRC_3 had the values 0.20, 0.50, and 0.80, respectively, for δ. For example, a δ value of 0.20 gives 20 percent importance to question content similarity and 80 percent importance to relationship similarity. We used the best results of the three combinations of the content parameter δ to proceed with improved quadripartite graph-based cluster analysis.

Finally, to validate cluster analysis using multiple approaches, we extended the improved quadripartite graph-based cluster analysis (IQR) to three different types of hierarchical clustering algorithms. We used the complete link algorithm (IQR_{CL}), average link algorithm (IQR_{AL}), and Ward's algorithm (IQR_W) (Balijepally et al., 2011). We used agglomerative cluster analysis as the baseline for content based clustering (AC). We used bipartite cluster analysis as the baseline for relationship based clustering (BR). Table 2 provides the results we obtained for the eleven different types of cluster analysis, which we discuss in Section 6.

We also established the validity of the cluster solution by ensuring that the clusters represented the actual population. SQA services build a bourgeoning amount of content in the form of questions, answers, ratings, reviews, and user profiles. Of the various SQA services available, we used Yahoo! Answers as the dataset for this study for three reasons: popularity, richness in metadata, and collective wisdom. According to the statistics that Hitwise (2013) reports, Yahoo! Answers held eighth position in the number of visits to social networking sites. Recent years have seen Yahoo! Answers' content rise significantly due to several reasons, such as its availability, ease of use for creating and sharing content, and the increasing number of people turning to collaboration (Chua & Banerjee, 2015), which results in an expanding information repository that holds immense potential for both social and market research. Additionally, using the representative dataset from the website's four most frequently used domains (arts and humanities, business and finance, computers and internet, and science and mathematics) helped ensure the external validity of the clusters.

5 Evaluation

Evaluating the artifact is an important stage. We tested the nine different combinations of algorithms and assessed their performance. Because we did not know the categories, we used the objective functions of the clustering algorithms to evaluate the algorithms. Researchers most often use precision, recall, and f-measure to perform such an evaluation (Xu et al., 2007); as such, we used them as the performance-evaluation metrics for this study.

Precision is a measure popular in information retrieval and is defined as the ratio of the number of relevant items retrieved and the total number of items retrieved. This metric is important because it measures the level of noise in the similar questions identified. For question clustering, we considered precision as the ratio of the number of similar questions to the total number of questions in a cluster (Leung et al., 2008). We calculated precision by examining 100 sample clusters to see if the questions in the clusters were actually similar (Wen et al., 2002). We then computed the overall precision as the average precision of all 100 question clusters.

Recall is another performance metric in information retrieval. Recall is the ratio of the number of relevant items retrieved to the total number of relevant items in the collection (Wen et al., 2002). However, for clustering questions, one measures recall as the ratio of the number of similar questions in the current cluster to the total number of all similar questions for a question set (Leung et al., 2008). Following this definition, calculating recall is complex because no standard clusters or classes were available. Hence, we used an alternate measure of recall, known as normalized recall, as Wen et al. (2002) propose. Normalized recall is the ratio of the number of questions judged as correctly clustered in the 100 sample clusters for a particular threshold to the maximum number of questions judged as correctly clustered in the 100 sample clusters across all thresholds. Wen et al. (2002) and Ray, Goh, and Foo (2006) clearly illustrated the use of normalized recall. The number of correctly clustered questions in 100 selected clusters equals the total number of questions in the 100-sample question clusters multiplied by the average precision. One computes the number of questions in the 100 selected question clusters by multiplying the average cluster size by 100. Further, to strike an even balance between precision and recall, we used the harmonic mean of precision and recall known as Van Rijsbergen's f-measure (1979).

As no predefined categories against which to judge the validity of the clusters existed, we compared clusters with reference to external knowledge using judgments made by two human evaluators. We needed two evaluators because judging the relatedness of questions is subjective and room existed for personal biases. We eliminated these biases to a great extent by taking an aggregate of the two evaluators' appraisals. Previous studies also used evaluators for relevance and quality judgments (Suryanto, Lim, Sun, & Chiang, 2009). We recruited two evaluators who had masters in information systems degrees and would work on the project for a nominal fee. They evaluated the clusters independently. Each cluster given for evaluation had two or more questions. We asked the evaluators to identify the questions in a cluster that had the same meaning. Since, in some cases, it is difficult to correctly understand the user's intention, evaluators made the best guess. We calculated the degree of agreement between evaluators by adopting Cohen's kappa statistic (Cohen, 1960), which resulted in a Cohen kappa value of 0.82 as the measure of agreement between the two evaluators for their manual evaluation of the clusters. Finally, we calculated the metrics for each evaluator's sample and averaged the two figures obtained to get the final performance metrics. Table 2 gives the performance results for eleven different cluster analyses.

In summary, we used two baseline techniques (AC and BR). We used three quadripartite graph-based cluster analyses with the relationship similarities (QR_1, QR_2, and QR_3) to identify the best combination of α, β, and γ. We used three quadripartite graph-based cluster analyses with the combined similarities (QRC_1, QRC_2, and QRC_3) to identify the best values for δ. Finally, we used three combinations of the improved cluster analyses (IQR_{CL}, IQR_{AL}, and IQR_W) to validate the algorithm with complete link algorithm, average link algorithm, and Ward's algorithm. The results reported in Table 2 clearly demonstrate that quadripartite graph-based cluster analysis with the relationship similarity ($QR_{1)}$, the content similarity ($QRC_{1)}$, and the improved cluster analyses (IQR_{CL}, IQR_W) performed better than the baseline clustering techniques.

Table 1. Different Clustering Algorithm Results

Algorithm	Threshold	Precision	Recall	F-measure
AC	0.9	73.3	41.2	52.8
	0.7	51.7	62.3	56.5
	0.5	48.3	78	59.7
BR	0.9	69.5	49.6	57.9
	0.7	56.2	54.7	55.4
	0.5	43.2	73.2	54.3
QR_1	**0.9**	**100**	**47.2**	**64.1**
	0.7	**65.2**	**59.3**	**62.1**
	0.5	**64.3**	**70.2**	**67.1**
QR_2	0.9	0	0	0
	0.7	100	33.3	33.3
	0.5	100	20	50
QR_3	0.9	100	20	33.3
	0.7	100	20	33.3
	0.5	100	33.3	50
QRC_1	**0.9**	**100**	**56.7**	**72.4**
	0.7	**70**	**68**	**69**
	0.5	**62.3**	**80.5**	**70.2**
QRC_2	0.9	0	0	0
	0.7	0	0	0
	0.5	100	33.3	50
QRC_3	0.9	0	0	0
	0.7	100	20	33.3
	0.5	100	20	33.3
IQR_{CL}	**0.9**	**96.8**	**63.8**	**76.9**
	0.7	**83.4**	**88.8**	**86.1**
	0.5	**70.1**	**95.4**	**80.8**
IQR_{AL}	0.9	92.9	31.4	46.9
	0.7	87.0	44.0	58.4
	0.5	82.3	56.1	66.7
IQR_W	**0.9**	**87.7**	**57.4**	**69.4**
	0.7	**77.9**	**82.7**	**80.2**
	0.5	**75.7**	**94.7**	**84.2**

Legend:
AC: Agglomerative cluster analysis using the content similarity $sim_{QC}(Q_i,Q_j)$
BR: Bipartite graph-based cluster analysis using similarity based on the question and answer concept relationship
QR_1: Quadripartite graph-based cluster analysis using the relationship similarity $sim_{QR}(Q_i,Q_j)$ with α = 0.80, β = 0.10, and γ = 0.10
QR_2: Quadripartite graph-based cluster analysis using the relationship similarity $sim_{QR}(Q_i,Q_j)$ with α = 0.33, β = 0.33, and γ = 0.33
QR_3: Quadripartite graph-based cluster analysis using the relationship similarity $sim_{QR}(Q_i,Q_j)$ with α = 0.20, β = 0.40, and γ = 0.40
QRC_1: Quadripartite graph-based cluster analysis using the combined similarity $sim_{QRC}(Q_i,Q_j)$ with δ = 0.20, α = 0.80, β = 0.10, and γ = 0.10
QRC_2: Quadripartite graph-based cluster analysis using the combined similarity $sim_{QRC}(Q_i,Q_j)$ with δ = 0.50, α = 0.80, β = 0.10, and γ = 0.10
QRC_3: Quadripartite graph-based cluster analysis using the combined similarity $sim_{QRC}(Q_i,Q_j)$ with δ = 0.80, α = 0.80, β = 0.10, and γ = 0.10
IQR_{CL}: Improved quadripartite graph-based complete link cluster analysis using the relationship similarity $sim_{QR}(Q_i,Q_j)$ with α = 0.80, β = 0.10, and γ = 0.10
IQR_{AL}: Improved quadripartite graph-based average link cluster analysis using the relationship similarity $sim_{QR}(Q_i,Q_j)$ with α = 0.80, β = 0.10, and γ = 0.10
IQR_W: Improved quadripartite graph-based ward cluster analysis using the relationship similarity $sim_{QR}(Q_i,Q_j)$ with α = 0.80, β = 0.10, and γ = 0.10

6 Discussion

In this section, we discuss how our results shed new light in understanding the influence of the quadripartite graph-based cluster analysis in identifying similar questions.

6.1 Analysis of the Relationship Similarity Measure

We found that the quadripartite graph-based cluster analysis with the relationship similarity measure *(*QR_1, QRC_1, IQR_{Cl}, IQR_{AL} and IQR_W*)* overcame the lexical mismatch in questions by incorporating a question-answer-asker-answerer relationship similarity measure. The algorithm performed better than both agglomerative cluster analysis using content similarity and bipartite cluster analysis using relationship similarity. In effect, this finding helped verify our assumption that the intricate network of relationships among questions, answer concepts, askers, and answers played a vital role in overcoming the lexical gap in identifying similar questions. It also validated that the weights of 80 percent for answers and 10 percent each for askers and answerers gave the best results for quadripartite graph-based cluster analysis with the relationship similarity measure *(*QR_1). However, the performance of the quadripartite graph-based cluster analysis was very low when we gave equal weight to answers, askers, and answerers in QR_2 or gave a higher weight to askers and answerers than answer in QR_3. Thus, we concluded that the main reason for QR_1 to perform better than QR_2 and QR_3 was that the answers were rich in words that were common. Hence, the answers played a vital role in clustering similar questions.

6.2 Analysis of Users: Askers and Answerers

On further analysis, we found that, in the dataset of the 5733 unique questions, unique askers contributed 5309 questions, and these questions attracted responses from 3211 unique answerers. We examined the questions associated with top askers and answerers to identify how they contributed to Yahoo! Answers.

The top asker asked 14 questions on insurance. The questions that the top asker asked were all related. For example, “Can anybody tell me about the term insurance plan?” and “Why should I choose an insurance plan instead of another savings plan?”. It seemed that, if the asker received a satisfactory answer and marked it as the best answer, the asker would probe the same topic with more questions. We found this same trend for the second top asker who asked nine questions on insurance, which clarifies our assumption that similar askers post similar questions.

We also found that some answerers very actively answered questions. The top answerer answered 131 unique questions on history and answered questions from 121 unique askers. The questions ranged from topics related to world wars, Adolf Hitler, Al-Qaeda, and the Renaissance. Hence, we needed to more specifically examine each answer’s content. The second top answerer answered 54 questions in three different domains: advertising and marketing, insurance, and the Internet. Among the 54 questions answered, 48 questions concerned advertising and marketing. The second top answerer gave separate answers to 54 different askers on various questions related to the same topic. In this case, it meant that answerers played a role in identifying similar questions, but we needed to depend on not only the answerer but also the answer itself to identify similar questions.

Finally, we found that 116 users both asked and answered questions. In other words, 116 users served as asker and answerer. Among them, a user who asked 14 questions about insurance answered four questions on insurance. A user asked nine questions about insurance and answered three questions. The user who answered 19 questions on insurance asked one question in the same domain. Another user who asked six questions on advertising and marketing answered seven questions in the same domain.

Hence, from analyzing the roles that askers and answerers played, we found that askers and answerers do play an important role in identifying similar questions. For future research, we recommend including a user’s profile for tracing that user’s expertise (Bian et al., 2009) to improve clustering, enhance retrieval and to reduce noise and outliers.

6.3 Analysis of Content – Questions and Answers

Quadripartite graph-based cluster analysis with the relationship similarity measure (QR_1, QRC_1, IQR_{CL}, IQR_{AL}, and IQR_W) used the lexical content in answers to overcome the lexical mismatch in questions. Among the variables used for the relationship similarity measure, we found the answer component to be more significant than askers and answerers. An in-depth analysis revealed that 5733 questions received 5581 unique answers. The most repeated answer was on guidelines for downloading videos from YouTube, and it came from three different answerers who answered seven, ten and two times, respectively. The second most repeated answer was for questions on Myspace. We also found a pair of questions with no lexical match but that the analysis clustered them because they had common answers and answerers. The analysis also clustered the questions “How did the great society and the Reagan revolution seek to change America?” and “What are some examples of freedom that Ronald did during his presidency?” as similar. In this case, the questions had no overlap in phrases. Hence, in such a situation, a common answer and answerer aided in clustering the questions together with no common words.

Few answers were general in nature; for example, “Please refer to Google”, which led to clustering unrelated questions based on similar answers. We could see that using answers alone to identify similar questions was misleading, particularly with general answers. Quadripartite graph-based cluster analysis with the combined similarity measure was able to weed out the issues in identifying similar questions based solely on relationship. Also, QRC_1 with a 20 percent weight for question content similarity and 80 percent for relationship similarity performed better than QRC_2 and QRC_3. Equal weight for question content and relationship resulted in 100 percent precision for the lowest threshold; however, the recall was very low. Moreover, the 90 percent threshold resulted in zero clusters. The main reason for why QRC_1 performed better than QRC_2 and QRC_3 was that the answers, askers, and answerers had more impact than the question content itself.

6.4 Analysis of Algorithms

Based on comparing various algorithms overall, QR_1 achieved 100 percent precision for the 90 percent threshold. However, QRC_1 achieved higher precision than QR_1 at all threshold levels. Improved quadripartite graph-based cluster analysis based on relationship improved recall and maintained precision, which resulted in the best range of performance.

On the other hand, we found that the complexity of clustering considered space and time. The space complexity for the algorithm was directly related to the number of questions (n_q), number of unique answer concepts (n_{ac}), number of unique askers (n_{as}), and number of unique answerers (n_{an}). The memory allocation required for calculating similarity depended on the number of corresponding vertices. To reduce space complexity, we considered only unique askers and answerers. Moreover, we extracted unique answer concepts only from the best answers. We also found that the time complexity of the algorithm was related to the number of question-question similarity measure evaluations. The quadripartite graph-based cluster analysis completed one cycle by merging similar questions, similar answer concepts, similar askers, and similar answerers in turns. However, improved quadripartite graph-based cluster analysis completed one cycle by merging a similar question and its respective answers, askers, and answerers and, thus, reduced the time complexity.

We tested the complete link algorithm, average link algorithm, and Ward’s algorithm to ensure validity and reliability of the improved quadripartite graph-based cluster analysis. To check the reliability of the cluster solutions based on the subcategories of Yahoo! Answers, we identified the subcategories for each question that the analysis clustered at the 90 percent threshold. Interestingly, we found that all three algorithms (IQR_{CL}, IQR_{AL}, IQR_W) obtained 100 percent reliability in clustering the questions in the same subcategory at the 90 percent threshold. Overall, as an average for all threshold levels, we found that the average link algorithm showed 100 percent reliability, followed by Ward’s algorithm (89%), and the complete link algorithm (79%). Thus, improved quadripartite graph-based cluster analysis improved not only the time and space complexity but also the clustering results, which the f-measure in Table 2 shows.

7 Conclusion

As Fayard and DeSanctis (2008) indicate, intimate relationships between content and users are possible online as a community develops. By fostering the relationship between user-generated content and associated users, we followed the IS design science research approach to design and evaluate a quadripartite graph-based cluster analysis approach for clustering similar questions. Hence, our study sheds light on the richness of the relationship built in SQA services and their complex reality by identifying similar questions. Thus, this study contributes to the literature by forging a four-way link among questions, answers, askers, and answerers so that we can better identify similar questions.

Identifying similar questions allow us to retrieve answers associated with similar questions, which reduces the associated time lag in waiting for other users to answer questions. It also adds value to existing services by better allowing us to reuse user-generated questions and answers collected in SQA services. For SQA service designers, our findings offer implications for fine-tuning their answer retrieval and recommendation service by harnessing the relationships between content and users. Thus, as the SQA services maintain enormous sets of resolved questions, graph-based cluster analysis is an effective method to revitalize the information contained in their archives and to serve the needs of their users as promptly as possible.

This paper has three limitations. First, the assumption that similar askers will ask similar questions underpinning the proposed quadripartite clustering algorithm appeared to hold well for the current dataset drawn from four categories of Yahoo! Answers. However, we do not know how the algorithm would perform in settings involving more diverse categories/subcategories of questions/answers. Because askers can post questions on diverse topics, one should compare the performance of the quadripartite clustering solution with the tripartite clustering algorithm that includes only the questions, answers, and answerers.

Second, as identifying similar questions enhances the reuse of answers, we need to trace the quality of the answers. Identifying the quality of the answers depends on not only the accuracy of the content (Blooma, Chua, & Goh, 2012) but also the effectiveness of the answer (Chua & Banerjee, 2015). We need to explore the intimate relationships between questions, answers, askers and answerers to identify the factors that affect the quality of the answers (Bian et al., 2009). Hence, future research should investigate features that affect the quality of the answers to improve the retrieval of answers, an area that has hitherto received little attention.

Third, we need to design the clustering algorithm to dynamically collect data. An algorithm that can dynamically collect data is highly important today because SQA services play a vital role for businesses to treat the market as a conversation between themselves and customers (Chen, Chiang, & Storey 2012). Improved quadripartite graph-based cluster analysis is the first step toward improving the time complexity while considering the intimate relationships between content and users involved in SQA services. Future research should focus on integrating scalable cluster analysis to cater to the ever-increasing amounts of questions and answers in various business- and healthcare-oriented SQA services (Oh, 2012; Blooma & Wickramasinghe, 2014).

Thus, as the quadripartite network is mutable in the context of identifying similar answers or users, the generalizability of the algorithm will make it more valuable. We demonstrate how graph-based cluster analysis can solve the complexity of user-generated content in social media. One could extend our study to collaborative applications that help users seek, share, and recommend information and build up social relationships. By modelling the quadripartite network into social networking services, we can leverage big data and integrated Web applications for catering to customer queries.

Acknowledgements

We would like to acknowledge the 2013 College of Business Early Career Researcher Publication Scheme from RMIT University for providing a financial support of US$4,420 at the initial stage of this manuscript. We would also like to thank the senior editor, Jeffery Parsons, and the anonymous reviewers for their helpful comments on the three versions of this manuscript.

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734-749.

Amrit, C., Wijnhoven, F., & Beckers, D., (2015). Information waste on the World Wide Web and combating the clutter. In *Proceedings of European Conference of Information Systems.*

Arnott, D., & Pervan, G. (2012). Design science in decision support systems research: An assessment using the Hevner, March, Park, and Ram Guidelines. *Journal of the Association for Information Systems, 13*(11), 923-949.

Arazy, O., & Kopak, R. (2011). On the measurability of information quality. *Journal of the American Society for Information Science and Technology*, *62*, 89-99.

Balijepally, V., Mangalaraj, G., & Iyengar, K. (2011). Are we wielding this hammer correctly? A reflective review of the application of cluster analysis in Information Systems research. *Journal of the Association for Information Systems, 12*(5), 375-413.

Beeferman, D., & Berger, A. (2000). Agglomerative clustering of a search engine query log. *In Proceedings of the 6th ACM International Conference on Knowledge Discovery and Data Mining* (pp. 406-416). NY: ACM.

Bian, J., Liu, Y., Zhou, D., Agichtein, E., & Zha, H. (2009). Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceeding of the 18thIinternational Conference on World Wide Web* (pp. 51-60). NY: ACM.

Blooma, M. J., Chua, A. Y. K., & Goh, D. H. (2012). Predictors of high-quality answers. *Online Information Review*, 36(3), 383-400.

Blooma, M. J., Kurian, J. C., Chua, A. Y. K., Goh, D. H., & Lien, N. G. (2013). Social question answering: Analyzing knowledge, cognitive processes and social dimensions of micro-collaborations. *Computers and Education, 69,* 109-120.

Blooma, M. J., & Kurian, J. C. (2012). Clustering similar questions in social question answering services. In *Proceedings of 16th Pacific Asia Conference on Information Systems.*

Blooma, M. J., Chua, A. Y. K., & Goh, D. H. (2011). Quadripartite graph-based clustering of questions. In *Proceedings of the 8th International Conference on Information Technology: New Generations*. NY: IEEE Computer Society.

Blooma, M. J., & Wickramasinghe, N. (2014). Healthcare social question answering: Concept mapping and cluster analysis based on graph theory. In *Proceedings of the Australasian Conference on Information Systems.*

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., & Hwang, D. U. (2006). Complex networks: Structure and dynamics. *Physics Reports, 424*(4-5), 175-308.

Chua, A. Y. K., & Banerjee, S. (2015). Measuring the effectiveness of answers in Yahoo! Answers. *Online Information Review, 39*(1), 104-118.

Chen, H., Chiang, R. H., & Storey, V. C. (2012), Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, *36*(4), 1165-1188.

Chen, Y., & Fonseca, F. (2003). A bipartite graph co-clustering approach to ontology mapping. In *Proceedings of the Workshop of Semantic Web Technologies for Searching and Retrieving Scientific Data*.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Education Psychology Measurement, 20*(1), 3-46.

Dhillon, I. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data mining* (pp. 269-274). NY: ACM.

Fayard, A.L., & DeSanctis, G. (2008) Kiosks, clubs and neighborhoods: the language games of online forums, *Journal of the Association for Information Systems 9* (10/11), 677-705.

Gregor, S., & Hevner, A.R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly, 37*(2), 337-355.

Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association of Information Systems, 8*(5), 312-335.

Grujic, J., Mitrovic, M., & Tadic, B. (2009). Mixing patterns and communities on bipartite graphs on Web-based social interactions. In *Proceedings of 16th International Conference on Digital Signal Processing* (pp. 1-8). NY: IEEE Computer Society.

Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., & Tatham, R. L. (2006). *Multivariate data analysis.* (6th ed.). Upper Saddle River, NJ: Pearson Prentice-Hall.

Harper, F. M., Moy, D., & Konstan, J. A. (2009). Facts or friends? Distinguishing informational and conversational questions in social Q&A sites. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems* (pp. 759-768). NY: ACM.

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.

Hitwise. (2013). *Marketing suite*. Retrieved from http://www.experian.com/marketing-services/online-trends-social-media.html

Joo, K. H., & Lee, S. (2005). An incremental document clustering algorithm based on a hierarchical agglomerative approach. In G. Chakraborty (Ed.), *Distributed computing and Internet technology* (LNCS 3816, pp. 321-332). Berlin: Springer.

Kim, J., & Park, H. (2008). Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In *Proceedings of the 18th IEEE International Conference on Data mining* (pp. 353-362). NY: IEEE Computer Society.

Lambiotte, R., & Ausloos, M. (2005). Uncovering collective listening habits and music genres in bipartite networks. *Physics Review E, 72*(6), 66107.

Lambiotte, R., & Ausloos, M. (2006). Collaborative tagging as a tripartite network. In V. N. Alexandrov, G. D. Albada, P. M. A. Sloot, & J. Dongarra (Eds.)*, Computational science* (LNCS 3993, pp. 1114-1117). Berlin: Springer.

Leung, K. W. T., Ng, W., & Lee, D. L. (2008). Personalized concept-based clustering of search engine queries. *IEEE Transactions on Knowledge and Data Engineering, 20*(11), 1505-1518.

Li, X. Y., Yuan, J. S., & Jing, Y. W. (2007). An efficient user access pattern clustering algorithm. In *Proceedings of the 6th International Conference on Machine Learning and Cybernetics* (pp. 4109-4112). NY: IEEE Computer Society.

Liu, Y., Li, S., Cao, Y., Lin, C. Y., Han, D., & Yu, Y. (2008). Understanding and summarizing answers in community-based question answering services. In *Proceedings of the 22nd International Conference on Computational Linguistics* (pp. 497-504). NJ: Association of Computational Linguistics.

Lu, C., Chen, X., & Park, E. K. (2009). Exploit the tripartite network of social tagging for Web clustering. In D. W. K. Cheung, I. Song, W. W. Chu, X. Hu, & J. Lin (Eds.), *Proceedings of the 2009 ACM International Conference on Information and Knowledge Management* (pp. 1545-1548). NY: ACM.

Myers, M., & Venable, J. (2014). A set of ethical principles for design science research in information systems. *Information & Management, 51*(6), 801-809.

Oh, S. (2012). The characteristics and motivations of health answerers in social Q&A. *Journal of the American Society for Information Science and Technology*, *63*(3), 543-557.

Punj, G., & Stewart, D. W. (1983). Cluster analysis in marketing research: Review and suggestions for application. *Journal of Marketing Research, 20*(2), 134-148.

Ray, C. S., Goh, D. H., & Foo, S. (2006). The effect of lexical relationships on the quality of query clusters. In *Proceedings of the 9th International Conference on Asian Digital Libraries* (LNCS 4312, pp. 223-233).

Rege, M., Dong, M., & Fotouhi, F. (2006). Co-clustering documents and words using bipartite isoperimetric graph partitioning. In *Proceedings of the 6th IEEE International Conference on Data Mining International* (pp. 5320541). NY: IEEE Computer Society.

Rege, M., Dong, M., & Hua, J. (2008). Graph theoretical framework for simultaneously integrating visual and textual features for efficient web image clustering. In J. Huai, R. Chen, H. W. Hon, Y. Liu, W. Y. Ma, A. Tomkins, & X. Zhang (Eds.), *Proceeding of the 17th international conference on World Wide Web* (pp. 317-326). NY: ACM.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval.* NY: McGraw-Hill.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, *1*(1), 27-64.

Shachaf, P. (2010). Social reference: Toward a unifying theory. *Library & Information Science Research*, 32(1), 66-76.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. In *Proceedings of the 6th ACM International Conference on Knowledge Discovery and Data Mining* (pp. 109-111). NY: ACM.

Suryanto, M. A., Lim, E. P., Sun, A., & Chiang, R. H. L. (2009). Quality-aware collaborative question answering: Methods and evaluation. In *Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval* (pp. 142-151). NY: ACM.

Takeuchi, K. (2008). Extraction of verb synonyms using co-clustering approach. In *Proceedings of the 2nd International Symposium on Universal Communication* (pp. 173-178). NY: IEEE computer Society.

Tamura, A., Takamura, H., & Okumura, M. (2005). Classification of multiple-sentence questions. In L. P. Kaelbling & A. Saffiotti (Eds.), In *Proceedings of the 2nd International Joint Conference on Natural Language Processing* (LNAI 3651, pp. 426-437). Berlin: Springer.

Tombros, A., Villa, R., & Van Rijsbergen, C. J. (2002). The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing and Management: An International Journal, 38*(4), 559-582.

Van Rijsbergen, C.J. (1979). *Information Retrieval.* Butterworths, London, 1979.

Walter, B., Bala, K., Kulkarni, M., & Pingali, K. (2008). Fast agglomerative clustering for rendering. In *Proceedings of the IEEE Symposium on Interactive Ray Tracing* (pp. 81-86). NY: IEEE Computer Society.

Wen, J., Nie, J., & Zhang, H. (2002). Query clustering using user logs. *ACM Transactions on Information Systems, 20*(1), 59-81.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J., & Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge Information Systems*, *14*(1), 1-37.

Xu, J., Wang, G. A., Li, J., & Chau, M. (2007), Complex problem solving: Identity matching based on social contextual information. *Journal of the Association for Information Systems, 8*(10), 525-545.

Xu, Y., & Yin, J. (2015). Collaborative recommendation with user generated content. *Engineering Applications of Artificial Intelligence, 45,* 281-294.

Zhao, Y., & Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the 11th International Conference on Information and Knowledge Management* (pp. 515-524). NY: ACM.

About the Authors

Blooma Mohan John is Assistant Professor at the Faculty of Business Government and Law, University of Canberra. She has a PhD in Information Systems from Nanyang Technological University. Her research interests are in text mining, social question answering, learning analytics and health informatics. She has published various academic articles including journal papers, book chapters and conference proceedings in these areas.

Alton Y.K. Chua is Associate Chair (Research) and Associate Professor at the Wee Kim Wee School of Communication and Information, Nanyang Technological University. His research interests lie in information and knowledge management, and in particular, social informatics. He has published in excess of 150 academic articles including journal papers and conference proceedings in these areas. He also serves on the editorial board of a number of journals including the Journal of Information Science and the Journal of Information and Knowledge Management.

Dion Goh has a PhD in computer science. He is currently Associate Professor with Nanyang Technological University (Singapore) where is also the Director of the Masters of Information Systems program in the Wee Kim Wee School of Communication and Information. His major areas of research are in social media perceptions and practices, gamification techniques for shaping user perceptions and motivating behaviour, as well as mobile information sharing and seeking.

Nilmini Wickramasinghe is the inaugural Professor-Director of Health Informatics Management and Professor of Health Informatics at Deakin University's faculty of health. Professor Wickramasinghe is an internationally recognised scholar who researches and teaches within health informatics with a particular focus on developing suitable models, strategies and techniques grounded in various management disciplines to facilitate more effective design, development and implementation of IS/IT solutions to effect superior, patient centric healthcare delivery. She is well published with more than 320 referred scholarly articles, more than 12 books, numerous book chapters, an encyclopaedia, and a well-established funded research track record.