# Unsupervised deep semantic and logical analysis for identification of solution posts from community answers

by

Niraj Kumar, Kannan Srinathan, Vasudeva Varma

in

*Int. J. Information and Decision Sciences*

Report No: IIIT/TR/2016/-1

Centre for Search and Information Extraction Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2016

# Unsupervised deep semantic and logical analysis for identification of solution posts from community answers

Niraj Kumar* and Kannan Srinathan

Department – Center for Security Theory and Algorithmic Research,
IIIT-Hyderabad,
Hyderabad-500032, India
Email: niraj_kumar@research.iiit.ac.in1
Email: srinathan@iiit.ac.in
*Corresponding author

Vasudeva Varma

Department – Search and Information Extraction Lab 'SIEL',
IIIT-Hyderabad,
Hyderabad-500032, India
Email: vv@iiit.ac.in

**Abstract:** These days' discussion forums provide dependable solutions to the problems related to multiple domains and areas. However, due to the presence of huge amount of less-informative/inappropriate posts, the identification of the appropriate problem-solution pairs has become a challenging task. The emergence of a variety of topics, domains and areas has made the task of manual labelling of the problem solution-post pairs a very costly and time consuming task. To solve these issues, we concentrate on deep semantic and logical relation between terms. For this, we introduce a novel semantic correlation graph to represent the text. The proposed representation helps us in the identification of topical and semantic relation between terms at a fine grain level. Next, we apply the improved version of personalised pagerank using random walk with restarts. The main aim is to improve the rank score of terms having direct or indirect relation with terms in the given question. Finally, we introduce the use of the node overlapping version of GAAC to find the actual span of answer text. Our experimental results show that the devised system performs better than the existing unsupervised systems.

**Keywords:** normalised pointwise mutual information; NPMI; semantic correlation graph; personalised pagerank algorithm; random walk with restart; RWR; community question answering; semantic relatedness; group average agglomerative clustering.

**Reference** to this paper should be made as follows: Kumar, N., Srinathan, K. and Varma, V. (2016) 'Unsupervised deep semantic and logical analysis for identification of solution posts from community answers', *Int. J. Information and Decision Sciences*, Vol. 8, No. 2, pp.153–178.

**Biographical notes:** Niraj Kumar is a PhD research scholar from IIIT-Hyderabad. His areas of interests are: text mining, data mining, web mining, social networking and machine learning. He has worked in several core areas text mining like: keyphrase extraction, document summarisation,

summarisation evaluation, automatic evaluation of descriptive answers, automatic question answering, plagiarism detection and recommendation systems.

Kannan Srinathan is an Assistant Professor of Computer Science and Engineering Department at IIIT-Hyderabad, Hyderabad, India. He is associated with the security theory and algorithmic research lab 'CSTAR' of IIIT-Hyderabad. His areas of interests are (including but not limited to) cryptography, security, graph theory and distributed computing.

Vasudeva Varma is a Professor of Computer Science and Engineering at IIIT-Hyderabad, Hyderabad, India. He is the Head of Search and Information Extraction Lab 'SIEL' of IIIT-Hyderabad. His research interests are in the broad areas of information retrieval, extraction and access. More specifically: social media analysis, cross language information access, summarisation and semantic search. He also works in the areas of cloud computing and reuse in software engineering.

## 1 Introduction

Discussion forums play a very important role in a lot of different areas. For example, Yahoo-answer (https://in.answers.yahoo.com/) provides online discussion and solutions related to more than 26 categories (each having multiple sub categories), Stack overflow (http://stackoverflow.com/) provides an online discussion forum for 40,000 categories, related to the technical domain and so on. A lot of product and service companies have their discussion forums. Thus, all these discussion forums contain a huge variety and diversity in the area of discussion. All these discussion forums contain questions from registered users and discussion posts. Here the number of posts in discussion threads vary and depend upon the popularity of that topic.

According to Deepak and Visweswariah (2014) and Kolodner (1992), extracting problem-solution pairs from forums enables the usage of such knowledge in knowledge reuse frameworks such as case-based reasoning that use problem-solution pairs as raw material.

Cong et al. (2008) had proposed the first unsupervised system to deal with this problem. They employ a graph propagation method that prioritises posts that are:

a more similar to the problem post,

b more similar to other posts

c authored by a more authoritative user, to be labelled as solution posts.

On the other hand the latest method (Deepak and Visweswariah, 2014) models and harnesses lexical correlations using translation models in the company of unigram language models that are used to characterise reply posts and formulate a clustering-based EM approach for solution identification. However, we believe that without proper utilisation of the knowledge resources and deep analysis of semantic-logical correlation and relatedness, it is tough to explore a lot of relation between terms, which may be very useful in answering the given question. It will clear from the following discussions.

## *1.1 Problem definition*

The following contains important issues/problems, which should be given high attention in the process of getting the solution posts for any given discussion thread.

**Table 1** A sample question and solution post

| | |
|---|---|
| *Question:* | Why does a snake flick-out its tongue?? |
| *Solution post:* | ‘A *snake smells* by using its *forked tongue* to collect *airborne particles* then passing them to the *Jacobson's organ* or the *Vomeronasal organ* in the *mouth* for *examination*. The *fork* in the *tongue* gives the *snake* a sort of *directional sense* of smell. The part of the *body* which is in direct contact with the surface of the ground is very sensitive to vibration, thus a snake is able to sense other animals approaching.’ |

- *Problem-1:* can we give equal weight/importance to all non-stopword terms of the given question?

  *Observation*: we believe that the majority of the times, non-stopword terms in the given question may have different topical depth/strength. e.g., from the question given in Table 1, it is clear that ‘snake’ shows high topical strength than non-stopword terms ‘flick-out’ and ‘tongue’. While, both terms (i.e., ‘flick-out’ and ‘tongue’) seem to have nearly similar topical strength. Considering such facts in the automatic identification of the most suitable answer post may be very important.

- *Problem-2:* can we neglect the role of terms (in posts), which do not have a direct match with terms in the given question?

  *Observation*: from the solution post given in Table 1, it is clear that a lot of important bold faced words play a very important role in the answer of the given question. Some of them have no exact match with non-stopword terms given in the question. For example, the bold faced terms like: *airborne particles*, *Jacobson's organ*, *directional sense* and *smell* etc., have no exact/direct match with terms in the given question like: *snake*, *flick out* or *tongue*. Without using the knowledge resources and deep semantic-logical analysis, it is tough to identify such relations.

- *Problem-3:* it is clear from the above discussion that some terms play a very important role in answering the question. Even, some of them do not have a match with the terms in the given question. Now, the questions are:

  1 ‘What type of relations exists between such terms?’

  2 ‘What are the usefulness of such relations in answering the question?’

  *Observation:* from the example given in Table 1, it is clear that, there exist two types of relations between the terms of the given question and terms in the solution post. The first one has been discussed earlier, i.e., in the observation section of ‘problem-2’. The second one is the semantic-logical relation between the terms of the annotated answer. For example, the relation between terms, like: *airborne particles*, *Jacobson's organ*, *directional sense* and *smell*, etc. This second relation helps us in identifying all terms, which do not have a direct match with the terms of the given question, but directly/indirectly plays a very important role in answering the question. We can exploit both types of relations to identify and validate the suitable answer of the question.

- *Problem-4:* can we identify the required amount of information content for a good answer?

  *Observation*: it is possible to collect all keywords, which covers the complete information about any valid/quality answer. For example, bold faced terms in the answer, given in Table 1, represent the good set of useful terms and cover all the important information required for the answer. Identifying such terms will be very useful in the effective identification of the answer of any given question.

From the above discussion, it is clear that, a fine-grained analysis of role of terms in the posts and utilising the importance of terms in the given question can give some better insight towards the solution of this problem.

### *1.2 Motivation*

Based on the above discussed facts and observations, we concentrate on:

1 the development of a novel way to represent the text

2 improvement in the ranking scheme

3 finally, development of a novel solution post extraction approach.

The following contains the important steps to solve the problem and motivation behind them.

1 First of all, we develop a proper text representation system. The main goal is to effectively represent the semantic correlation and relatedness between:

   a terms of the given question and answer posts

   b terms of the answer posts (if any).

   To achieve this, we introduce the use of semantic correlation graph.

2 Second, we give high ranks to all the terms which have sufficient semantic and logical importance and very useful in answering the question. To achieve this goal, we introduce the use of an improved version of personalised pagerank using random walk with restarts (RWRs).

3 Finally, we collect a set of all terms which capture the complete information required to answer the given question. To achieve this goal, we introduce the use of node overlapping-based group average agglomerative clustering algorithm (GAAC).

Thus, the devised system uses the three major techniques to achieve the desired goal, i.e.,

1 semantic correlation graph

2 an improved version of personalised pagerank using RWR

3 node overlapping-based GAAC.

The following contains the background technical discussions and the motivation behind the use of the proposed techniques.

*Semantic correlation graph*. The way to construct the graph for any given text plays a very important role in information extraction. For example, when we prepare the word graph of text by using the adjacent word pairs (as used in Kumar et al., 2013a, 2013b), then it gives the semantic and information flow related information about the adjacent words of the given text. Thus, it captures the limited information about the text (w.r.t., the requirement of the question-answering task). For example, a traditional bigram graph-based representation of the example given in Table 1 can be represented as:

**Figure 1** Traditional bigram graph representation of text

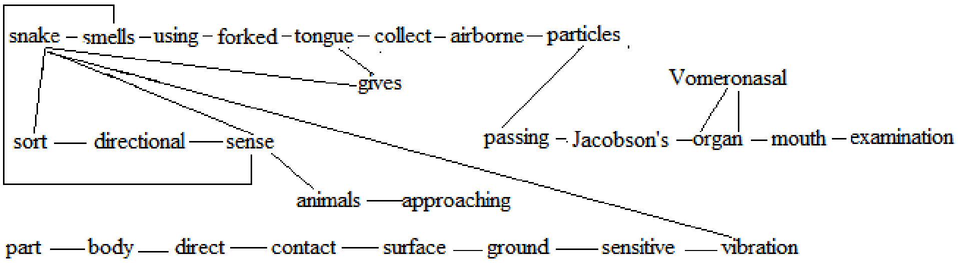


Figure 1 shows the graph-based representation of the text ‘a *snake smells* by using its *forked tongue* to collect *airborne particles* …’. Here some relations are missing, i.e., the relation between ‘snake and airborne particles’, and ‘snake and tongue’, etc. To effectively answer any question, we require a more elaborate relation between the terms of answer posts and questions. This is possible, if we extract a sufficient amount of domain specific nearest neighbour documents from Wikipedia and use it to extract semantic correlations between:

1 important terms in the answer posts

2 non-stopword terms in the given question and answer post.

We use this intuition in preparing the graph and named it as a semantic correlation graph.

*Modified version of personalised pagerank, using RWRs*. This algorithm is based on the RWRs process, so the random surfer will uniformly choose random nodes for restarting with a possibility $P_{|U|}$ during the random walk process.

$$RR(Y) = (1-\beta)\left(\sum_{V\in adj(U)} \frac{RR(V)}{L(V)}\right) + (\beta)P_{|U|} \tag{1}$$

Here, $RR(U)$ represents the personalised pagerank of node ‘$U$’ using RWRs. $\beta$ represents the restart probability ($0 \leq \beta \leq 1$), which determines, how often it restarts at the set of root nodes in ‘$R$’. We define a vector $P_R$ of prior probabilities $P_R = \{p_1,\ldots,p_{|V|}\}$ such that the probabilities sum to 1 and where $p_{|V|}$ denotes the relative importance (or ‘prior bias’) we attach to node ‘$V$’. If we give equal importance to each page/node of the set related to the set of prior nodes, then, $P_{|U|}$ (relative importance or ‘prior bias’, we attach to node to the given set of nodes and represented as ‘$R$’) can be calculated as:

$$P_{|U|} = \begin{cases} 1/|R| & \text{For } 'U' \in R \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

Among various proximity measures, RWR is widely adopted because of its ability to consider the global structure of the whole network. Other merit of a RWR is that it can model the multifaceted relationship between two nodes (Gori et al., 2007). The personalised pagerank using RWR makes each node get a higher ranking score, if the node is more closely related to the nodes that exist in the query (i.e., non-stopword question-terms in the current scenario). Actually, Lee et al. (2011), uses the personalised pagerank using RWRs for 'multi-dimensional recommendation'.

However, to meet our requirements, we have made some improvements in the 'personalised pagerank'. The main reasons are:

1 we cannot treat each non-stopword terms in the given question equally

2 non-stopword terms of the given candidate solution posts will have different semantic correlation strength with the terms of the given question.

So, we add both facts in the personalised pagerank, using RWRs. This scheme gives high rank to terms, which have high role in the answer of the given question (whether it matches with the terms in the given question or not).

*Node overlapping-based GAAC:* among three major agglomerative clustering algorithms, i.e., single-link, complete-link and average-link clustering. Single-link clustering can lead to elongated clusters. Complete-link clustering is highly affected by outliers. Average-link clustering is a compromise between the two extremes, which generally avoids both problems. This is the main reason of use of GAAC for clustering the graph.

GAAC uses average similarity across all pairs within the merged cluster to measure the similarity between the clusters.

However, to get the overlapping clusters, we have made a change in assigning nodes to the clusters. If the distances/similarities from node '$P$' to cluster '$C1$' and to cluster '$C2$' are close enough then put '$P$' in both clusters. The cluster formation scheme is same as given in two-phase algorithm discussed by Dash and Liu (2001). The average similarity between two clusters (say: $c_i$ and $c_j$) can be computed as:

$$sim\left(c_i, c_j\right) = \frac{1}{\left|c_i \cup c_j\right|\left(\left|c_i \cup c_j\right| - 1\right)} \sum_{\vec{x}\in\left(c_i \cup c_j\right)} \sum_{\vec{y}\in\left(c_i \cup c_j\right):\vec{y}\neq\vec{x}} sim(\vec{x}, \vec{y}) \tag{3}$$

where

$sim(\vec{x}, \vec{y})$ count of co-occurring words in $\vec{x}$ and $\vec{y}$.

We use 0.4 as the similarity threshold. This is the best performing threshold used in all experimental evaluations.

*Justifying the use of GAAC:* in any given candidate text, most of the important terms, responsible to answer the question, contain very high level of topical and semantic closeness. We can exploit their semantic and sometimes direct connections to cluster them. As discussed earlier, with the help of semantic correlation graph and modified version of personalised pagerank using RWRs we can give high scores to all such terms. Now, by using the GAAC, we cluster all such terms, which have nearly similar score and may have some connectivity between them (if represented as a semantic correlation graph of text). We believe that, term cluster obtained by using appropriate overlapping-based clustering algorithms can easily capture all the required information to answer the given question.

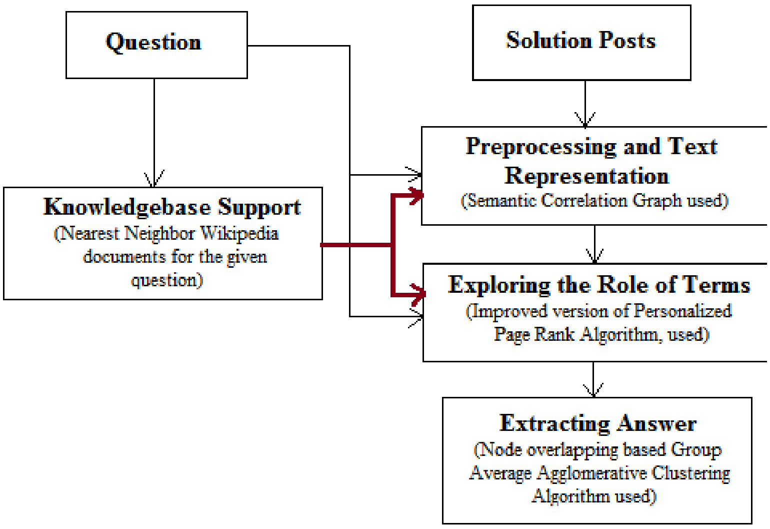
## 1.3 Framework of the devised system

We divide the entire system into three operational phases.

In the first phase (i.e., pre-processing and text representation), we pre-process all solution posts and use the graph-based representation (i.e., introduced the semantic correlation graph) to represent the all solution posts (candidate answer text).

In the second phase, we explore the role of non-stopword terms in the given solution posts to identify the appropriate solution posts for the given question. See, 'exploring the role of terms', Figure 2.

**Figure 2** Framework of the devised system (see online version for colours)


Question
Solution Posts
Preprocessing and Text Representation
(Semantic Correlation Graph used)
Knowledgebase Support
(Nearest Neighbor Wikipedia documents for the given question)
Exploring the Role of Terms
(Improved version of Personalized Page Rank Algorithm, used)
Extracting Answer
(Node overlapping based Group Average Agglomerative Clustering Algorithm used)

Finally, in the third phase, we extract the answer (the appropriate solution post for the given question). See, 'extracting answer', Figure 2.

## 1.4 Our contribution

Based on the above discussion, our contribution towards the development of the entire system can be summarised as follows:

- *Preparing semantic correlation graph:* we introduce a novel semantic correlation graph-based representation of text for automatic solution post identification from online discussion threads. The proposed representation helps us in the identification of topical and semantic relation between terms at a fine grain level and very useful in identification of correct solution post for any given question in discussion threads. To achieve this we use the nearest neighbour Wikipedia documents.
- *Improved version of personalise pagerank algorithm, calculated by using a RWRs:* we introduce the modified version of personalise pagerank algorithm, calculated by using a RWRs. The proposed scheme exploits:
  1. the importance of non-stopword question terms
  2. semantic correlation of terms obtained by using 'semantic correlation graph'.

It gives high rank to the terms, which plays a very important role in answering the given question.

- *Extracting the answer:* to identify the set of terms, which captures all important information for the answer of the given question, we introduce the use of partially overlapping-based group average agglomerative clustering (GAAC) algorithm. The proposed scheme, clusters and gives high rank to all the terms, which have an important role in answering the given question. With the help of the highest ranked term cluster, we can easily extract the most appropriate answer (answer/solution post) for the given question (discussion thread).

### *1.5 Paper overview*

In Section 2, we briefly discuss the related works of this area. In Section 3, we extract the nearest neighbour Wikipedia articles for the given question and calculate the normalised pointwise mutual information (NPMI) score for the selected set of word pairs. In Section 4, we prepare a semantic correlation graph by using all the solution posts related to the given question (i.e., candidate answer text). In Section 5, we identify the role/importance of non-stopword terms in the given candidate answer text, w.r.t., the given question. In Section 6, we identify a highly representative set of terms to answer the given question and finally, by using it, we identify the most appropriate answer for the given question. In Section 7, we present the pseudo code. In Section 8, we present the experimental evaluation of the devised system.

## 2 Related work

Based on the techniques applied, we can group the previous work (related to the problem addressed in this paper) into the following categories.

1 *Unsupervised techniques:* a very few unsupervised techniques exist in this area. For example: Cong et al. (2008), propose sequential patterns-based classification method to detect questions in a forum thread and a graph-based propagation method to detect answers for questions in the same thread. (Deepak and Visweswariah, 2014), uses the translation models and language models to exploit lexical correlations and solution post character respectively.

2 *Semi-supervised techniques:* Catherine et al. (2013), propose two semi-supervised methods for extracting answers from the discussions, which utilise the large amount of unlabeled data available, alongside a very small training set to obtain improved accuracies. They show that it is possible to boost the performance by introducing a related, but parallel task of identifying acknowledgments to the answers.

3 *Supervised techniques:* a lot of supervised techniques exist in this area. For example: Wang et al. (2009), treated questions and their answers as relational data and proposed an analogical reasoning-based method to identify correct answers. They assume that there are various types of linkages which attach answers to their questions and used a Bayesian logistic regression model for link prediction. In order to bridge the lexical gap, they leverage a supporting q-a set whose questions are

relevant to the new question and which contain only high-quality answers. This supporting set together with the logistic regression model is used to evaluate:

a how probably a new q-a pair has the same type of linkages as those in the supporting set

b how strong it is.

The candidate answer that has the strongest link to the new question is assumed as the best answer that semantically answers the question. Qu and Liu (2011), propose a two-step approach to classify online forum threads according to their informativeness in terms of question answering. They use statistical models to first categorise posts inside a thread. Then, a variety of features including post level information and other meta-data information is used to classify the thread. Ding et al. (2008), propose a general framework based on conditional random fields (CRFs) to detect the contexts and answers of questions from forum threads. They improve the basic framework by skip-chain CRFs and 2D CRFs to better accommodate the features of forums for better performance. Kim et al. (2010), introduce the tasks of:

a post classification, based on a novel dialogue act tag set

b link classification.

They also introduce three feature sets (structural features, post context features and semantic features) and experiment with three discriminative learners (maximum entropy, SVM-HMM and CRF). They achieve above baseline results for both dialogue act and link classification with interesting divergences in which, feature sets perform well over the two subtasks and go on to perform a preliminary investigation of the interaction between post tagging and linking. Hong and Davison (2009), show that the use of N-grams and the combination of several non-content features can improve the performance of detecting question-related threads in discussion boards. They show that the number of posts a user starts and the number of replies produced and their positions are two crucial factors in determining potential answers. They show that relevance-based retrieval methods would not be effective in tackling the problem of finding possible answers, but the performance can be improved by combining with non-content features while we treat retrieval scores as features. Using classification results, they are able to design a simple ranking scheme that outperforms previous approaches when retrieving potential answers from discussion boards.

However, none of the discussed method claims the fine grain analysis of the role of terms in answering the question (as discussed in the previous section).

## 3 Extracting nearest neighbour Wikipedia articles and calculating NPMI score

To calculate the semantic correlation between word pairs, we use the *NPMI* (Bouma, 2009) score, calculated by using nearest neighbour Wikipedia documents. We use Wikipedia link structure (anchor text-based) for faster calculation of the nearest neighbour Wikipedia documents. The following contains the necessary steps, used to calculate the *NPMI* score of bigrams/word-pairs.

### 3.1 Identifying Wikipedia anchor-text communities

Wikipedia has the well-organised anchor text link structure and most of the Wikipedia anchor texts have a corresponding descriptive article, which in turn contains other anchor texts (related to the context of the topic) in its body text. We use this link structure in the preparation of the graph. In this scheme, we consider every anchor text as a node of the graph. Finally, we apply the edge betweenness strategy [as applied in Girvan and Newman, (2002)] to identify the anchor text communities. This scheme is same as discussed by Kumar et al. (2010).

### 3.2 Identifying document's categories and indexing

We uniquely map Wikipedia documents with respect to each of the identified anchor text community. For this, we use the title-based matching. We also check the category information for any category-title related disambiguation. Thus, we create a Wikipedia document category for each of the identified Wikipedia anchor text community. Next, we apply the *LUCENE*-based (http://lucene.apache.org/) indexing for each of the identified document categories. We use these indexes in the calculation of *NPMI* scores of bigrams/word-pairs.

### 3.3 Calculating NPMI

From the list of Wikipedia anchor-text communities, we extract all Wikipedia anchor text communities, which show high cosine similarity with the given candidate answers text. Next, we select the related Wikipedia document's category and merge the indexes of all extracted document categories by using *LUCENE* (if the given candidate document matches with more than one Wikipedia anchor text communities). As, we use pre-calculated indexes of Wikipedia documents, related to each of the identified Wikipedia anchor text community, so merging the required number of identified indexes is a lightweight process. Finally, we use the *LUCENE*-based indexes to calculate *NPMI*. For this, we use the text window of size 20 words (one to two sentences approx.). The equation for *NPMI* (Bouma, 2009) can be given as:

$$NPMI\left(t_i, t_j\right) = \begin{cases} -1 & \text{if } p\left(t_i, t_j\right) = 0 \\ \dfrac{\log p\left(t_i\right) + \log p\left(t_j\right)}{\log p\left(t_i, t_j\right)} - 1 & \text{otherwise} \end{cases} \tag{4}$$

where

$p(t_i, t_j)$ is the joint probability and can be calculated by counting the number of observations of words $t_i$ and $t_j$ in a window of size 20 words

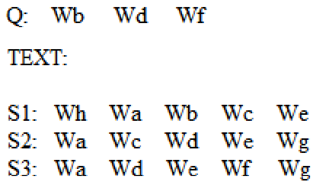
$p(t_i)$ probability of occurrence of $t_i$ in Wikipedia documents and so on.

*Analysis:* as, NPMI gives the values in the range [–1, +1]. The word-pairs having NPMI score greater than zero, generally show tight and observable semantic strength between them. After a lot of observations, we have found that word pair having NPMI scores greater than zero can be considered as semantically important and useful for the calculation. We use this as an important constraint in the entire calculation.

## 4 Pre-processing and text representation

Our pre-processing step includes the removal of noisy terms, stemming and sentence boundary detection. We use porter stemmer (http://tartarus.org/martin/PorterStemmer/) for stemming. We append all solution posts in a single file (we call it as a candidate answer text). See Figure 3, for sample question and candidate answer text.

**Figure 3** Question and candidate answer text



Q: Wb Wd Wf

TEXT:

S1: Wh Wa Wb Wc We
S2: Wa Wc Wd We Wg
S3: Wa Wd We Wf Wg

Notes: ‘Q’ represents a question,
‘S1’, ‘S2’ and ‘S3’ represents the candidate answer text,
‘Wa’, ‘Wb’, ‘Wc’, ‘Wd’ … and so on represent distinct words

Our text representation system uses a semantic correlation graph to represent the text. For this, we use the:

1 *NPMI* score between the terms of the given question and the terms of the candidate text

2 *NPMI* score between the terms of the candidate answer text.

*Identifying the semantic correlation:* to identify the semantic correlation between word pairs, we represent the given question and candidate answer text in the form of a bipartite graph. Where, the left side of the node contains the question terms and right side of the nodes contain the distinct terms from the candidate answer text. Formally, we can define it as bipartite graph $G = (L, R, E)$, where, ‘*L*’ represent the left set of nodes, ‘*R*’ represent the right set of nodes, and ‘*E*’ represents the edge of the graph.

We add the link between nodes of ‘*L*’ and ‘*R*’ (i.e., bipartite edge);

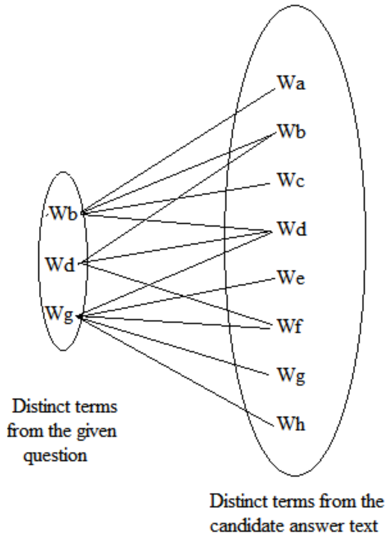
1 if they match with each other (see the bipartite edge ‘*Wb Wb*’ in Figure 4)

2 if they show some semantic strength (if they do not match, see the bipartite edge ‘*Wa Wb*’ in Figure 4).

In the second case, we check the semantic strength between them. For this, first of all we calculate the *NPMI* score obtained by using the nearest neighbour documents. We add an edge ‘*E*’ between any two such nodes, which belong to ‘*L*’ and ‘*R*’, if its *NPMI* score is greater than zero (see ‘analysis’, Sub-section 3.3). For example, the bipartite graph given in Figure 4. The edge ‘*Wa Wb*’, ‘*Wb Wc*’ shows that *NPMI*(*Wa*, *Wb*) and *NPMI*(*Wb*, *Wc*) is greater than zero. In the first case (i.e., nodes between ‘*L*’ and ‘*R*’ match with each other), we take NPMI score as ‘one’ (highest score).

We use a sample question and candidate answer text (as given in Figure 3) to demonstrate the identification of the semantic correlation and finally the formation of the semantic correlation graph of the text.

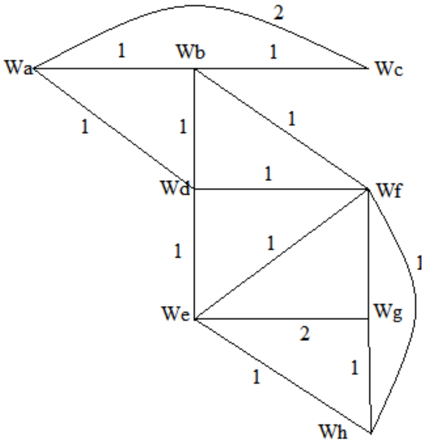
**Figure 4** Bipartite graph, based on the semantic relation between non-stopwords terms of the given question and candidate answer


Wa
Wb
Wc
Wd
We
Wf
Wg
Wh
Wb
Wd
Wg
Distinct terms from the given question
Distinct terms from the candidate answer text

Note: See Figure 3, for question and candidate text/candidate answer text

*Preparing semantic correlation word graph of text:* by using the bipartite graph based on the semantic relation between non-stopword terms, we extract all semantic co-relation pairs, i.e., all pairs of nodes (i.e., distinct non-stopwords terms from candidate answer) which share the common node from the other side of bipartite graph (i.e., contains non-stopwords question terms). For example, in Figure 4, nodes 'Wa' and 'Wb' are connected to the common node 'Wb' through bipartite edges. So, we consider 'Wa' and 'Wb' as semantic correlation pairs. Similarly, ('Wa', 'Wc') and ('Wb', 'Wc'), etc., are semantic correlation pairs. We extract all such semantic correlation pairs and prepare the semantic correlation graph.

**Figure 5** Semantic correlation graph, for candidate answer text



Note: Constructed by using the bipartite graph given in Figure 4

Formally, the semantic correlation graph can be given as $G = (V, E)$, where the distinct non-stopwords terms from the given candidate text are considered as vertices, i.e., $V = V_a$, $V_b$, $V_c,\ldots,V_n\}$, where, $V_a$, $V_b$, etc., are distinct terms from the given candidate text.

We add the edge between any two vertexes $V_i$ and $V_j$, if there exist bipartite edges, '$V_iV_k$' and '$V_jV_k$', where $\{V_i, V_j \in R\}$ and $\{V_k \in L\}$ and $NPMI(V_i, V_j) > 0$. Here, '$L$' represents the left set of nodes and '$R$' represents the right set of nodes. For example, Figure 5 represents the semantic correlation graph, constructed by using the bipartite graph given in Figure 4.

Now, to calculate the edge weight between $V_i$ and $V_j$, i.e., $EdgeWt(V_i, V_j)$, we use the semantic strength between $V_i$ and $V_j$, and cardinality of $V_i$ and $V_j$.

$$EdgeWt\left(V_i, V_j\right) = NPMI\left(V_i, V_j\right) \times \min\left(count\left(V_i\right), count\left(V_j\right)\right) \tag{5}$$

where

$count(V_i)$ count of occurrences of term (node) $V_i$ in the candidate document,

$\min(count(V_i), count(V_j))$ minimum of the count of occurrences of $V_i$ and $V_j$ in the candidate text.

This minimum count gives the maximum possible co-occurrences of nodes $V_i$ and $V_j$ (i.e., the main reason behind the selection of minimum count). We use this graph in the calculation of role of terms in the actual answer of the given question.

## 5 Exploring role/importance of terms

We use a modified version of personalised pagerank using RWR, on the semantic word graph of text. This technique gives high rank to all those terms, which plays a very important role in answering the question.

### *5.1 Modified version of personalise pagerank using RWRs*

For this, we use an adjacency matrix to represent the semantic correlation word graph of text. We convert it into a row stochastic matrix, by normalising the row sums of the corresponding transition matrix to one. Finally, we calculate the prior/bias probability $P_{|U|}$ of every non-stopword term in the given question. The algorithm is based on a RWR process, so the random surfer will choose a random node for restarting with a possibility $P_{|U|}$ during the random walk process. Let '$R$' represents the set of non-stopword terms in the given question and $P_{|U|}$ denotes the prior probability, we attach to node '$U$'.

$$P_{|U|} = \begin{cases} \dfrac{TfIdf(U)}{TfIdf\_Sum} & \text{For } 'U' \in R \\ 0 & \text{Otherwise} \end{cases} \tag{6}$$

where

*TfIdf*(*U*) *Tf-Idf* score of term '*U*' w.r.t., nearest neighbour Wikipedia articles (see Section 3, for calculating the nearest neighbour Wikipedia articles for any given question)

*TfIdf_Sum* represents the sum of *Tf-Idf* score of all non-stopword terms in the given question (i.e., present in the set '*R*').

The following contains a modified version of the personalise pagerank (Haveliwala et al., 2003) by using RWRs:

$$RR(U) = (1-\beta)\left( \sum_{V \in adj(U)} \frac{EdgeWt(U,V)}{\sum\limits_{W \in adj(V)} EdgeWt(W,V)} \times RR(V) \right) + \beta P_{|U|} \tag{7}$$

where

*RR*(*U*) personalise pagerank score of node '*U*', *obtained by using RWRs*

*EdgeWt*(*U*, *V*) edge weight of the edge between '*U*' and '*V*'

*adj*(*U*) the set of nodes, which are adjacent to node '*U*'

*adj*(*V*) the set of nodes, which are adjacent to node '*V*'

$\beta$ represents the restart probability ($0 \leq \beta \leq 1$), which determines, how often it restarts at the set of root nodes in '*R*'.

We use $\beta = 0.70$ (best performing setup, used in all experiments).

Actually, equation (7) adjusts the score with a personalise bias. By using this equation, we calculate the rank of all words in the given text file.

## *5.2 Advantages of adapting personalised pagerank using RWR*

There are several advantages of adapting the *personalised pagerank by using RWR*. First, we can take advantage of propagation and attenuation properties (Haveliwala et al., 2003). The propagation property is that the relatedness of the nodes propagates through following the links and the attenuation property is that the propagation strength decreases as the propagation goes further from the starting node (Page et al. 1998).

The personalise pagerank algorithm calculates the node authority value, but it adjusts the score with a personalise bias. It provides a proper control in the preference flow, in order to transfer high score values to the terms that are related to the query terms. We consider the non-stopwords terms given in question as query terms or as a user's preference and spread the user preferences through semantic graph.

By using these properties, we can also measure the relatedness between nodes which are not directly linked and nodes which are directly linked both. Thus the use of personalised pagerank using RWR on a semantic correlation graph gives higher rank to all the terms which are directly or indirectly play a very important role in the answer of the given question.

## 6 Extracting answer

We use personalise pagerank score of nodes to re-calculate the edge weight of the semantic correlation-based word graph of text (see Figure 5). This re-calculated edge weight is used to identify the similarity score of each adjacent connected node pair. We use these similarity scores to identify all graph communities by using node overlapping-based GAAC. Finally, we use the top ranked community to extract the most suitable answer post. The top ranked node overlapping-based GAAC contains the set of important terms for the most suitable answer post. For this we go through the following steps.

### *6.1 Recalculating edge weight and node similarity*

To recalculate the edge weight, we use the personalised pagerank score of nodes (see Section 4). The link weight of any edge $E = (V_i, V_j)$ can be given as:

$$W(V_i, V_j) = \left\{ \frac{Score(V_i)}{Degree(V_i)} + \frac{Score(V_i)}{Degree(V_i)} \right\} \times L_c(V_i, V_j)/2 \tag{8}$$

where

$W(V_i, V_j)$ Link weight of link between nodes $V_i$ and $V_j$

$Score(V_i)$ personalised pagerank score of node (word) $V_i$

$Score(V_j)$ personalised pagerank score of node (word) $V_j$

$Degree(V_i)$ degree of node (word) $V_i$

$Degree(V_j)$ degree of node (word) $V_j$

$L_C(V_i, V_j)$ count of number of links between nodes $V_i$ and $V_j$.

Note: we use the same link frequency, i.e., $L_C(V_i, V_j)$ as used in previous graph construction step (see Section 4). By using this scheme, we calculate the link weight of every edge of the graph.

*The similarity between two nodes:* to calculate the similarity between two nodes, we take the inverse of link weight [see equation (8) for calculation of link weight]. As, we believe that higher link weight will show more similarity between the adjacent nodes.

$$Sim(V_i, V_j) = {}^{1}\!/_{w(V_i, V_j)} \tag{9}$$

where $Sim(V_i, V_j)$= similarity between nodes $V_i$ and $V_j$.

### *6.2 Using graph clusters to identify the final answer post*

We use node overlapping-based GAAC to extract all word-clusters (node-clusters). We rank the identified word clusters by using the personalised pagerank score and use the highest ranked word cluster as the set of representative terms for the answer of the given question. However, to reduce the chances of lengthy candidate answers from getting higher rank, we use the score of words whose personalise pagerank score is greater than average score. Now, the word cluster importance score can be calculated as:

$$Score(C_i) = \sum_{W \in C_i} Score(W) \tag{10}$$

where

$C_i$ represents the $i^{\text{th}}$ word cluster

$Score(C_i)$ score of word cluster $C_i$

$Score(W)$ personalise pagerank score of word '$W$' in the given word cluster $C_i$.

In this calculation, we consider the score of only those words, whose score is above average. For the rest of the words, we take the score as zero.

Note: we sort all the identified word clusters in descending order of the calculated score.

*Identifying answer:* now, we select the solution post, which shows the highest cosine similarity with the highest ranked identified word cluster (i.e., the set of representative terms) as an answer for the given question.

## 7 Pseudo code

```
Input:     (1) Yahoo answer dataset, (2) Wikipedia document collection.
Output:    Question answer/solution-post pair.
Algorithm:
St1  For each of the discussion thread we do the following.
St2      We append all answer posts into a single file (call as candidate text/candidate answer
         text). Next, we prepare a semantic correlation word graph of the given candidate text
         (Section 4).
St3      We apply the improved version of personalize pagerank algorithm using random walk
         with restart on a word graph of text to calculate the rank of all words in the candidate
         text with respect to non-stopword terms of the given question (Section 5).
St4      We use the personalize pagerank score of words to re-calculate the link weight of the
         graph. Next, we use a node-overlapping version of the GAAC algorithm to cluster the
         graph nodes. We rank the extracted node (word) clusters in descending order of their
         weight (Subsection 6.1 to 6.3).
St5      We use the highest ranked term cluster to extract the solution thread/post for the given
         question.
St6  End For (Step St1).
```

## 8 Evaluation

We use Yahoo answer dataset (http://sourceforge.net/projects/yahoodataset/files/) to evaluate the performance of our devised system. This dataset contains total four categories and 26 sub-categories. The entire dataset contains total 11,123 discussion threads in the form of separate files. Out of 11,123 files, 3,300 files contain more than one answer posts, which are properly answered. Table 2 contains the details of the dataset:

**Table 2** Details of Yahoo answer dataset

| *Categories* | *(Sub-category, #files, #files_containing_more_than_one_answers and properly answered)* |
|---|---|
| NewCategoryIdentification | (NewCategoryIdentification, 5977, 2615) |
| Hardware | (Add-ons, 277, 105), (Desktops, 616, 102), (Laptops&Notebok, 384, 48), (Monitors, 285, 20), (Printers, 211, 11). |
| Internet | (Flickr, 199, 10), (Google, 213, 15), (Wikipedia, 201, 58), (Youtube, 198, 43) |
| Science | (Agriculture, 145, 39), (Astronomy&Space, 162, 55), (Biology, 144, 21), (Earthscience&Geology, 166, 26), (Geography, 149, 17), (Mathematics, 177, 20), (Medicine, 161, 18), (Physics, 139, 11), (Weather, 166, 18), (Zoology, 155, 34). |

*Answer annotation:* the answers given in the category ‘NewCategoryIdentification’ contains ‘chosen-answers’, annotated by users. Thus, for the category, ‘NewCategoryIdentification’, we consider the ‘chosen-answers’ as the gold standard and separate all such question answer pairs. For the rest of the categories, e.g., ‘hardware’, ‘internet’ and ‘science’, we apply a manual tagging to identify the solution or non-solution by deploying two expert annotators with an inter-annotator agreement (http://en.wikipedia.org/wiki/Cohen’s kappa) of 0.72. Actually, in 11 cases, two answers were suggested as valid answers by all annotators. So, in the evaluation process, if the extracted/identified solution post matches with any of the annotated answers, then we consider it as a right match for the extracted/identified solution post.

*Evaluation metrics:* we use precision, recall and F-measure score to evaluate the performance of our devised system and other competing systems.

## *8.1 Detail of systems used in evaluation process*

We use the following two graph-based baseline systems to compare the experimental results of our devised graph-based system.

### *8.1.1 System developed by Cong et al. (2008)*

We consider the graph-based unsupervised system developed by Cong et al. (2008) as one of the strongest baseline system to compare the performance of our devised system. For this, we implemented the system. The graph-based propagation method used by Cong et al. (2008), for automatic identification of solution posts, is a two-step process. In the first step, Cong et al. (2008) build graphs for candidate answers and then at the later step, they compute ranking scores of candidate answers using the graph.

*Building graphs*: given a question $q$, and the set $A_q$ of its candidate answers, Cong et al. (2008) build a weighted directed graph denoted as $(V, E)$ with weight function $w$: $E \rightarrow \Re$, where $V$ is the set of vertices and $E$ is the set of directed edges and $w(u \rightarrow v)$ is the weight associated with the edge $u \rightarrow v$. Each candidate answer in $A_q$ will corresponds to a vertice in $V$.

*Generating the edge set E:* given two candidate answers $a_o$ and $a_g$, the weight for edge $a_o \rightarrow a_g$ is computed by a linear interpolation of the three factors, namely the similarity computed from KL-divergence $KL(a_o|a_g)$, the distance of $a_g$ from $q$, i.e., $d(ag, q)$, and the authority of the author of $a_g$, denoted as $author(a_g)$.

$$w\left(a_o \rightarrow a_g\right) = \frac{1}{1 + KL\left(P\left(a_o\right)\middle|P\left(a_g\right)\right)} + \lambda_1 \frac{1}{d\left(a_g, q\right)} + \lambda_2\, author\left(a_g\right) \tag{11}$$

Cong et al. (2008), estimate the authority of an author in terms of the number of his replying posts and the number of threads initiated by him. Finally, the normalised weight $nw(a_o \rightarrow a_g)$ among all generators $g$ of $a_o$, $g \in G_{a_o}$ was calculated as:

$$nw\left(a_o \rightarrow a_g\right) = \lambda \frac{1}{\left|A_q\right|} + (1 - \lambda) \frac{w\left(a_o \rightarrow a_g\right)}{\sum_{g \in G_{a_o}} w\left(a_o \rightarrow a_g\right)} \tag{12}$$

*Applying propagation with initial score:* given a question q and its set Cq of candidate answer, the ranking score of a candidate answer a, $a \in C_q$ is computed recursively as follows.

$$P_r\left(q|a\right) = \lambda \frac{P_r\left(q|a\right)}{\sum_{t \in C_q} P_r\left(qt\right)} + (1 - \lambda) \sum_{v \in C_q} nw(v \rightarrow a) \times P_r\left(qv\right) \tag{13}$$

Note: as, per parameter setting used in Cong et al. (2008), we use $\lambda_1 = 0.8$ and $\lambda_2 = 0.05$ in equation (11), $\lambda = 0.01$ in equation (12) and $\lambda = 0.2$ in equation (13).

*Implementation issues and experimental settings:* we use an independent implementation (http://wekax.googlecode.com/svn/trunk/wekaUT/weka/core/metrics/KL.java) Bilenko (2006) of the Kullback-Leibler divergence (Kullback, 1997) as the similarity measure between posts; KL-divergence was seen to perform the best in the experiments reported by Cong et al. (2008). Finally, to reduce the chances of over-fitting problems, we did separate experiments by taking one sub-category at a time from the given categories. For example:

1 'NewCategoryIdentification' does not contain any defined sub-categories, so we apply five-fold cross validation and take the average of five trials of scores (i.e., precision, recall and F-measure) as the final score

2 next, as the category: 'hardware', contains five sub-categories, i.e., 'add-ons', 'desktops', 'laptops&notebok', 'monitors' and 'printers' (see Table 2).

We separately compute the solution posts for each of the given sub-categories. For example, we take sub-category: 'add-ons', apply five-fold cross validation and take the average of five trials of scores (i.e., precision, recall and F-measure) as the final score. We repeat the same process with other sub categories, i.e., 'desktops', 'laptops&notebok', 'monitors' and 'printers'. We take the average of precision, recall and F-measure scores of all sub-categories and present them as the average precision, recall and F-measure for the category, 'hardware'. We repeat the same process with all other categories, i.e., 'internet' and 'science'.

### *8.1.2 A system with the semantic graph-based concept*

The unsupervised system described by Kumar et al. (2013a) uses the traditional semantic graph and apply pagerank with prior-based scheme to rank the answer passages. We consider it as a basic system. We implemented the system for automatic identification of solution posts from community answers. For this, we apply a simple pre-processing step, which includes the removal of noisy terms, stemming and sentence boundary detection. We append all solution posts in a single file (as discussed in Section 4). We use the following three stages to represent the system.

*Graph construction:* Kumar et al. (2013a), constructs a word graph of sentences for the given text. It adds links between two words, if they co-occur together within a window of size two words (i.e., bigram) in the sentences of the candidate document. It also boosts the multi-word overlapping phrases in the word graph of sentences, which appear both in the given question and in the source document. For this it adds new links on word graph of sentences based on the number of times the matching bigram appeared in the question. To calculate the weight of any edge, it takes the product of their co-occurrence frequency and pointwise mutual information score calculated by using Wikipedia extended abstracts collection.

*Ranking:* Kumar et al. (2013a) applies ranking with prior to calculate the ranks of all words in the candidate document. For this it considers all non-stopword terms of the given question as prior or root for the ranking with prior. Similar to, Kumar et al. (2013a), we use back propagation probability 0.7 in the entire experimental evaluations (this is also the best performing setup in the current case).

*Answer (solution post) extraction:* to calculate the scores of posts (candidate answer), Kumar et al. (2013a), uses only top ranked terms, i.e., 25% of the top ranked words. To calculate the scores of a candidate answer, it adds the pagerank with prior scores of top ranked words, present in the answer. Finally, it ranks the solution posts in descending order of their weight. We consider the highest ranked posts (candidate answer) as the solution post.

*Implementation issues:* to implement the pagerank with prior method, used in the paper (Kumar et al., 2013a) we use JUNG-Library (http://jung.sourceforge.net/) and available source code. To calculate all semantic relations, we use LUCENE-based indexes obtained by using Dbpedia extended abstracts.

## *8.2 Experimental comparison results and discussion*

### *8.2.1 Experimental comparison*

We use two different systems (Sub-section 8.1), to compare the results generated by our unsupervised system. Table 3 contains the average precision, recall and F-measure score of on each of the categories. The detailed results on each of the sub-categories of the given categories of the dataset are given in Table 4.

**Table 3** Evaluation results

| *Dataset (category)* | *Technique* | *Precision* | *Recall* | *F-measure* |
|---|---|---|---|---|
| NewCategoryIdentification | *Our devised system* | 69.3 | 78.2 | 73.5 |
| | A system with semantic graph-based concept | 65.0 | 67.2 | 66.1 |
| | Unsupervised graph propagation (Cong et al., 2008) | 63.8 | 66.2 | 64.9 |
| Hardware | *Our devised system* | 70.9 | 76.3 | 73.5 |
| | A system with semantic graph-based concept | 62.7 | 64.1 | 63.4 |
| | Unsupervised graph propagation (Cong et al., 2008) | 61.1 | 68.4 | 64.5 |
| Internet | *Our devised system* | 72.9 | 76.6 | 74.7 |
| | A system with semantic graph-based concept | 61.0 | 64.7 | 62.7 |
| | Unsupervised graph propagation (Cong et al., 2008) | 60.0 | 64.5 | 62.1 |
| Science | *Our devised system* | 71.1 | 74.2 | 72.6 |
| | A system with semantic graph-based concept | 65.1 | 70.9 | 67.8 |
| | Unsupervised graph propagation (Cong et al., 2008) | 64.0 | 70.0 | 66.0 |

**Table 4** Detailed comparative results of three different systems at sub-category level

| *Sub-categories* | *(Precision, recall, F-measure)* | | |
|---|---|---|---|
| | *Our devised system* | *A system with semantic graph-based concept* | *Unsupervised graph propagation (Cong et al., 2008)* |
| | *Category: NewCategoryIdentification* | | |
| NewCategoryIdentification | (69.3, 78.2, 73.5) | (65.0, 67.2, 66.1) | (63.8, 66.2, 64.9) |
| | *Category: hardware* | | |
| Add-ons | (70.96, 76.36, 73.49) | (63.23, 64.43, 63.82) | (61.23, 68.43, 64.64) |
| Desktops | (70.70, 76.22, 73.22) | (63.0, 64.3, 63.64) | (61.0, 68.3, 64.4) |
| Laptops&Notebok | (70.56, 76.07, 73.07) | (60.88, 65.17, 62.96) | (60.88, 68.17, 64.27) |
| Monitors | (70.83, 76.36,73.36) | (61.12, 62.43,61.77) | (61.12, 68.43,64.52) |
| Printers | (71.09, 76.51, 73.63) | (63.34, 63.56, 63.45) | (61.34, 68.56, 64.76) |
| | *Category: internet* | | |
| Flickr | (72.91, 76.58, 74.66) | (61.07, 65.03, 62.71) | (60.11, 64.52, 62.22) |
| Google | (73.04, 76.72, 74.82) | (61.18, 65.15, 62.84) | (60.22, 64.64, 62.35) |
| Wikipedia | (72.78, 76.44, 74.52) | (60.96, 64.91, 62.59) | (60.0, 64.4, 62.1) |
| Youtube | (72.63, 76.30, 74.28) | (60.83, 64.79, 62.39) | (59.88, 64.28, 61.90) |

**Table 4** Detailed comparative results of three different systems at sub-category level (continued)

| *Sub-categories* | *(Precision, recall, F-measure)* | | |
|---|---|---|---|
| | *Our devised system* | *A system with semantic graph-based concept* | *Unsupervised graph propagation (Cong et al., 2008)* |
| | *Category: science* | | |
| Agriculture | (71.13, 74.21, 72.62) | (65.14, 70.93, 67.83) | (63.74, 69.82, 66.57) |
| Astronomy&Space | (71.69, 74.79, 73.24) | (65.65, 71.48, 68.41) | (64.24, 70.36, 67.14) |
| Biology | (70.77, 73.84, 72.26) | (64.81, 70.58, 67.49) | (63.42, 69.47, 66.24) |
| Earthscience&Geology | (70.98, 74.06, 72.48) | (65.00, 70.79, 67.70) | (63.61, 69.68, 66.44) |
| Geography | (71.89, 75.00, 73.45) | (65.83, 71.68, 68.60) | (64.42, 70.56, 67.33) |
| Mathematics | (67.14, 68.65, 67.94) | (61.49, 65.62, 63.46) | (60.17, 64.59, 62.28) |
| Medicine | (71.88, 75.00, 73.45) | (65.82, 71.68, 68.60) | (64.41, 70.56, 67.33) |
| Physics | (71.27, 74.36, 72.76) | (65.27, 71.07, 67.96) | (63.87, 69.96, 66.7) |
| Weather | (71.42, 74.58, 72.98) | (65.40, 71.22, 68.17) | (64.0, 70.1, 66.9) |
| Zoology | (71.55, 74.72, 73.11) | (65.53, 71.35, 68.29) | (64.12, 70.23, 67.02) |

### *8.2.2 Analyses of result*

1 As, per results given in Table 3, our devised system shows effective improvements over the baseline systems of this area. The stability in precision, recall and F-measure score on dataset related to four different categories (which include 26 different sub categories), shows the effectiveness of our devised system.

2 The results given in Table 3 and Table 4, show that the effective utilisation of three components (i.e., semantic correlation graph, the use of improved version of pagerank with prior using RWR and the use of GAAC in answer extraction) has highly improved the quality of the performance. Each of the discussed components has added some merits in the overall improvements in the quality of the devised system. For example:

   a the use of semantic correlation graph handles the text at finer grain level with respect to the other two systems, i.e., the system developed by Cong et al. (2008) and ‘a system with the semantic graph-based concept’ (discussed in Sub-section 8.1.2). The semantic correlation graph (as discussed in Section 4), considers only those terms (nodes) in entire calculation, which shows a certain level of semantic correlation with the non-stopword terms of the given question. Thus, it reduces the role of less-important and noisy terms in the entire calculation and hence improves the quality of the result.

   b RWR has provided a good relevance score between two nodes in a weighted graph and it has been successfully used in numerous settings, like automatic captioning of images, generalisations to the ‘connection sub-graphs’, personalised PageRank and many more (Tong et al., 2006). We have used the RWR with the improved version of pagerank with prior. This also helped us in improving the quality of the output of the devised system.

c The use of GAAC helps in filtering the set of all useful terms required to answer any given question. It also covers the logical and semantic similarity between them. Thus, it has helped in improving the quality of the results of the devised system.

3 Table 4, shows the detailed results, i.e., precision, recall and F-measure score for each of the subcategories of the given dataset. The experimental results show that our devised system shows stability in the performance on different variety of the dataset, w.r.t., the other competing systems. The relatively low performance on the subcategory: '*Mathematics*' is due to the presence of a lot of mathematical statements in the text. It is very tough for the semantic techniques used in the current work and traditional semantic technique (used in Kumar et al., 2013a) to effectively capture such statements.

4 The analysis of the individual contributions is discussed in the next sub-section.

## *8.3 Analysis of the individual contribution of the main components*

In this section, we test the individual contributions of three major components of the devised system, i.e.;

a 'semantic correlation graph'

b 'modified version of personalised pagerank, using RWRs'

c 'node overlapping-based GAAC for answer extraction'.

For this, we use the system, i.e., 'a system with the semantic graph-based concept' (discussed in Sub-section 8.1.2) as a basic system and replace the discussed modules with our proposed contributions and finally check the impact of these changes in the quality of the system. The results are given in Table 5. The following contains the experimental settings for the analysis of the individual contribution of the main components.

a *Semantic correlation graph*: to analyse the impact of semantic correlation graph, we only replace the 'graph construction' part of the system (as discussed in Sub-section 8.1.2) by our semantic correlation graph (as discussed in Section 4). We denote it as 'experimental setting-1'. This replacement helps us in evaluating the effect on the performance of the system having current 'semantic correlation graph' over the traditional semantic graph (Kumar et al., 2013a).

b *Modified version of personalised pagerank, using RWRs:* we replace the 'ranking' part of the system discussed in Sub-section 8.1.2, by our 'modified version of personalised pagerank, using RWRs' (Section 5). We denote it as 'experimental setting-2'. This replacement helps us in identifying the effect of current 'modified version of personalised pagerank, using RWRs' over the ranking with prior scheme used in Kumar et al. (2013a).

c *Node overlapping-based GAAC for answer extraction:* we replace the 'answer (solution post) extraction' part of the system discussed in Sub-section 8.1.2, by our proposed technique (i.e., 'node overlapping-based GAAC for answer extraction', see Sub-section 6.1 to 6.3). We denote it as 'experimental setting-3'. This replacement helps us in evaluating the effect on the performance of the system having current 'node overlapping-based GAAC for answer extraction' over the traditional direct threshold-based system (Kumar et al., 2013a).

**Table 5** Analysing the impact of different features in the quality of the devised system

| *Dataset (category)* | *Technique* | *Precision (±standard deviation)* | *Recall (±standard deviation)* | *F-measure (±standard deviation)* |
|---|---|---|---|---|
| NewCategoryIdentification | 'Experimental setting-1' | **67.6 (1.21)** | **73.6 (1.20)** | **70.5 (1.16)** |
| | 'Experimental setting-2' | 66.4 (1.30) | 70.2 (1.39) | 68.2 (1.35) |
| | 'Experimental setting-3' | 66.3 (1.34) | 69.9 (1.34) | 68.0 (1.34) |
| | A system with semantic graph-based concept | 65.0 (1.29) | 67.2(1.11) | 66.1(1.19) |
| Hardware | 'Experimental setting-1' | **68.0 (1.16)** | **71.5 (0.98)** | **69.7 (0.67)** |
| | 'Experimental setting-2' | 65.5 (1.13) | 68.8 (1.00) | 67.1 (1.06) |
| | 'Experimental setting-3' | 65.2 (1.37) | 67.7 (1.39) | 66.4 (1.38) |
| | A system with semantic graph-based concept | 62.7 (1.21) | 64.1 (1.04) | 63.4 (0.87) |
| Internet | 'Experimental setting-1' | **68.0 (0.13)** | **72.7 (0.08)** | **70.3 (0.16)** |
| | 'Experimental setting-2' | 64.9 (0.11) | 69.5 (0.09) | 67.1 (0.37) |
| | 'Experimental setting-3' | 63.3 (0.32) | 68.9 (0.41) | 65.9 (0.37) |
| | A system with semantic graph-based concept | 61.0 (0.15) | 64.7 (0.15) | 62.7 (0.19) |
| Science | 'Experimental setting-1' | **69.6 (1.30)** | **73.4 (1.36)** | **71.4 (1.34)** |
| | 'Experimental setting-2' | 68.1 (1.22) | 72.4 (1.24) | 70.2 (1.23) |
| | 'Experimental setting-3' | 67.7 (1.23) | 71.0 (1.38) | 69.3 (1.36) |
| | A system with semantic graph-based concept | 65.1 (1.28) | 70.9 (1.80) | 67.8 (1.52) |

*Analysis:* the results (given in Table 5), indicate that the use of the component, i.e.;

a 'semantic correlation graph' in 'experimental setting-1', has made the highest positive impact on the system. The low standard deviation in the result also indicates the robustness of this component. While, the other two components, i.e.;

b 'modified version of personalised pagerank, using RWRs'

c 'node overlapping-based GAAC for answer extraction', have shown the nearly similar improvements in the system (see the results under the heading, 'experimental setting-2' and 'experimental setting-3').

The role of each of these three core contributions is already discussed in Sub-section 8.2.2. The effective improvements in the performance of the basic system i.e., 'a system with the semantic graph-based concept' (discussed in Sub-section 8.1.2) after adding the proposed components, also justifies the use of the discussed techniques in designing the system. Here, the individual performances of all the core contributions are lower than the highest performance (see the performance of 'our devised system', Table 3, Table 4). However, a combination of these features highly improves the performance of our devised system (as discussed in Sub-section 8.2). Thus, we can justify our approach of developing the devised system by effectively utilising the above discussed techniques/core components.

### 8.4 Complexity analysis and related issues

The running time of the devised system depends upon the running time of the following key steps and techniques:

*Calculating NPMI score (used in Sub-section 4.4):* we consider the LUCENE-based indexing part as the offline process. Thus, we only concentrate on the calculation of NPMI score. In a traditional system having 2GB RAM and 2.2 GHz dual core processor, the devised system calculates the NPMI score of 7000 distinct bigrams.

*GAAC:* we use the binary heap to implement the GAAC algorithm and the time complexity of the system is $O(n^2 \log n)$, where, 'n' is the total number of nodes/words. (However, it is possible to improve it). We use the weka-based class implementation for *GAAC* (http://grepcode.com/file/repo1.maven.org/maven2/nz.ac.waikato.cms.weka/weka-dev/3.7.5/weka/clusterers/HierarchicalClusterer.java) with a slight change for node re-use, to get the node overlapping-based *GAAC*.

*RWR:* By using the algorithm, as discussed in Fujiwara et al. (2012), etc., we reduce the time complexity near to linear time in the terms of nodes of the graph. We add a bias factor to the random walk with researt class (source code) given in apache library (https://apache.googlesource.com/giraph/+/3d4f31343c3686435696e75ce88a75c9bffb024e/giraph-examples/src/main/java/org/apache/giraph/examples), to convert it into personalised pagerank using RWRs.

From the above discussion, It is clear that, our system can easily work as an online system to identify the solution post from online discussion forums.

## 9 Conclusions and scope

In this paper, we present a novel semantic correlation graph-based representation of text for automatic solution post identification from online discussion threads. We use this information to identify the most suitable group of terms, which contains all information, required to answer the given question. To identify the role of terms in answering the given question, we use the semantic correlation graph and improved version of personalised pagerank (using RWR). Finally, with the help of node overlapping-based GAAC and semantic ranking; we identify the most suitable answer post.

We are planning to explore this technique for:

1 guided summarisation task (where prior information is supplied to extract the most suitable summary sentences)

2 why-based question answering systems, etc.

Similarly, the technique used in identification of solution posts can be explored towards the automatic identification of the appropriate\optimal length guided summaries, descriptive answers and essays etc.

## References

Bouma, G. (2009) 'Normalized (pointwise) mutual information in collocation extraction', *Proceedings of The International Conference of the German Society for Computational Linguistics and Language Technology*, pp.31–40.

Bilenko, M.Y. (2006) *Learnable Similarity Functions and their Application to Record Linkage and Clustering*, PhD Thesis, Department of Computer Sciences, University of Texas at Austin, Vol. 67, No. 12, 136pp.

Catherine, R., Gangadharaiah, R., Visweswariah, K. and Raghu, D. (2013) 'Semi-supervised answer extraction from discussion forums', *IJCNLP*, Vol. 1, pp.1–9, ISBN 978-4-9907348-0-0.

Cong, G., Wang, L., Lin, C. Y., Song, Y. I. and Sun, Y. (2008) 'Finding question-answer pairs from online forums', *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development In Information Retrieval*, July, pp.467–474, ACM.

Dash, M. and Liu, H. (2001) 'Efficient hierarchical clustering algorithms using partially overlapping partitions', *PAKDD*, pp.495–506.

Deepak, P. and Visweswariah, K. (2014) 'Unsupervised solution post identification from discussion forums', *ACL*, Vol. 1, pp.155–164, ISBN 978-1-937284-72-5.

Ding, S., Cong, G., Lin, C. Y. and Zhu, X. (2008) 'Using conditional random fields to extract contexts and answers of questions from online forums', *ACL*, June, Vol. 8, pp.710–718, ISBN 978-1-932432-04-6.

Fujiwara, Y., Nakatsuji, M., Onizuka, M. and Kitsuregawa, M. (2012) 'Fast and exact top-k search for random walk with restart', *Proceedings of the VLDB Endowment*, Vol. 5, No. 5, pp.442–453.

Girvan, M. and Newman, M.E. (2002) 'Community structure in social and biological networks', *Proceedings of the National Academy of Sciences*, Vol. 99, No. 12, pp.7821–7826.

Gori, M., Pucci, A. and Roma, V. (2007) 'ItemRank: a random-walk based scoring algorithm for recommender engines', *IJCAI*, January, Vol. 7, pp.2766–2771.

Haveliwala, T., Kamvar, S. and Jeh, G. (2003) *An Analytical Comparison of Approaches to Personalizing Pagerank*, Technical Report, Stanford.

Hong, L. and Davison, B.D. (2009) 'A classification-based approach to question answering in discussion boards', *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development In Information Retrieval*, July, pp.171–178, ACM.

Kim, S.N., Wang, L. and Baldwin, T. (2010) 'Tagging and linking web forum posts', *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, July, pp.192–202, Association for Computational Linguistics, Chicago.

Kolodner, J.L. (1992) 'An introduction to case-based reasoning', *Artificial Intelligence Review*, Vol. 6, No. 1, pp.3–34.

Kullback, S. (1997) *Information Theory and Statistics*. Courier Corporation, Dover Publications, Inc., Mineola, New York.

Kumar, N., Gangadharaiah, R., Srinathan, K. and Varma, V. (2013a) *Exploring the Role of Logically Related Non-Question Phrases for Answering Why-Questions*, CoRR abs/1303. 7310.

Kumar, N., Srinathan, K. and Varma, V. (2013b) 'A knowledge induced graph-theoretical model for extract and abstract single document summarization', *CICLing*, Proceedings, Part II. Lecture Notes in Computer Science 7817, Springer, ISBN 978-3-642-37255-1, pp.408–423.

Kumar, N., Vemula, V.V.B., Srinathan, K. and Varma, V. (2010) 'Exploiting N-gram importance and Wikipedia based additional knowledge for improvements in GAAC based document clustering', *KDIR*, pp.182–187, Valencia, Spain, 25–28 October, SciTePress, ISBN 978-989-8425-28-7.

Lee, S., Song, S.I., Kahng, M., Lee, D. and Lee, S.G. (2011) 'Random walk based entity ranking on graph for multidimensional recommendation', *Proceedings of the Fifth ACM Conference on Recommender Systems*, October, pp.93–100, ACM.

Page, L., Brin, S., Motwani, R. and Winograd, T. (1998) *The Pagerank Citation Ranking: Bringing Order to the Web*, Technical report, Stanford digital library technologies project.

Qu, Z. and Liu, Y. (2011) 'Finding problem solving threads in online forum', *IJCNLP*, pp.1413–1417.

Tong, H., Faloutsos, C. and Pan, J.Y. (2006) 'Fast random walk with restart and its applications', *ICDM*, pp.613–622.

Wang, X.J., Tu, X., Feng, D. and Zhang, L. (2009) 'Ranking community answers by modeling question-answer relationships via analogical reasoning', *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, July, pp.179–186.