Journal of Biomedical Informatics 43 (2010) 1020–1035

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

A concept-driven biomedical knowledge extraction and visualization framework for conceptualization of text corpora

Jahiruddin [a], Muhammad Abulaish [a,*], Lipika Dey [b]

[a] Department of Computer Science, Jamia Millia Islamia (A Central University), New Delhi, India
[b] Innovation Labs, Tata Consultancy Services, New Delhi, India

ARTICLE INFO

Article history:
Received 18 May 2010
Available online 24 September 2010

Keywords:
Biological text mining
Biological relation extraction
Biomedical knowledge extraction and visualization
Semantic network
Biomedical query answering

ABSTRACT

A number of techniques such as information extraction, document classification, document clustering and information visualization have been developed to ease extraction and understanding of information embedded within text documents. However, knowledge that is embedded in natural language texts is difficult to extract using simple pattern matching techniques and most of these methods do not help users directly understand key concepts and their semantic relationships in document corpora, which are critical for capturing their conceptual structures. The problem arises due to the fact that most of the information is embedded within unstructured or semi-structured texts that computers can not interpret very easily. In this paper, we have presented a novel Biomedical Knowledge Extraction and Visualization framework, `BioKEVis` to identify key information components from biomedical text documents. The information components are centered on key concepts. `BioKEVis` applies linguistic analysis and Latent Semantic Analysis (LSA) to identify key concepts. The information component extraction principle is based on natural language processing techniques and semantic-based analysis. The system is also integrated with a biomedical named entity recognizer, ABNER, to tag genes, proteins and other entity names in the text. We have also presented a method for collating information extracted from multiple sources to generate semantic network. The network provides distinct user perspectives and allows navigation over documents with similar information components and is also used to provide a comprehensive view of the collection. The system stores the extracted information components in a structured repository which is integrated with a query-processing module to handle biomedical queries over text documents. We have also proposed a document ranking mechanism to present retrieved documents in order of their relevance to the user query.

1. Introduction

The number of text documents disseminating knowledge in biomedical field has gone up many folds as scientific publications and other forms of text-based data are produced at an unprecedented rate due to growing research activities in the recent past. Most scientific knowledge is registered in publications and other unstructured representations that make it difficult to use and to integrate the information with other biological data sources. Given that almost all current biomedical knowledge is published in scientific articles, researchers try to make use of this information. Consequently there is an increasing demand for automatic curation schemes to extract knowledge from scientific documents and store them in a structured form without which the assimilation of knowledge from this vast repository is becoming practically impossible. Knowledge discovery could be of major help in the discovery of indirect relationships, which might imply new scientific discoveries. Such new discoveries might provide hints for experts working on specific biological processes. While search engines provide an efficient way of accessing relevant information, the sheer volume of the information repository on the Web makes assimilation of this information a potential bottleneck in the way its consumption. One approach to overcome this difficulty could be to use intelligent techniques to collate the information extracted from various sources into a semantically related structure which can aid the user for visualization of the content at multiple levels of complexity. Such a visualizer provides a semantically integrated view of the underlying text repository in the form of a consolidated view of the concepts that are present in the collection, and their inter-relationships as derived from the collection along with their sources. The semantic net thus built can be presented to users at arbitrary levels of depth as desired.

Several disciplines including *information extraction, document classification, document clustering, and information visualization* have

* Corresponding author.
E-mail addresses: jahir.jmi@gmail.com (Jahiruddin), abulaish@ieee.org (M. Abulaish), lipika.dey@tcs.com (L. Dey).

1532-0464/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2010.09.008

been developed to ease extraction and understanding of information embedded in unstructured text documents [3–7]. However, knowledge that is embedded in natural language texts is difficult to extract using simple pattern matching. Although, techniques such as simple pattern matching can highlight relevant text passages from large abstract collection, generating new insights to future research is far more complex. Text mining has emerged as a hybrid discipline on the edges of the fields of information science, bioinformatics and computational linguistics which attempts to find hidden knowledge in the literature by exploring the structure of the knowledge network created using textual information [1,2,8].

In this paper, we have proposed the design of a novel biomedical knowledge extraction and visualization framework, BioKEVis, for conceptualization of document corpora and biomedical query answering. Conceptualization of document corpora here means representation and visualization of document corpora with a set of concepts and their relationships which can provide distinct user perspectives and allows navigation over documents with similar information components. BioKEVis applies Latent Semantic Analysis (LSA) to identify key concepts. Relationships among key concepts are extracted using natural language processing and semantic-based analysis. The information components are centered on key concepts and their relationships, and stored in structured form. The process of extracting relevant information components from text documents and automatic construction of structured knowledge bases is termed as curation which is very effective in managing online journal collections [15]. Schutz and Buitelaar [14] state that verbs play an important role in defining the context of concepts in a document. BioKEVis is designed to locate and characterize verbs within the vicinity of biological entities in a text, since these can represent biological relations that can help in establishing query context better. The verbs thus mined from documents are subjected to feasibility analysis and then characterized at concept level. We have shown that relation mining can yield significant information components from text whose information content is much more than entities.

Besides mining relational verbs and associated entities, the novelty of the system lies in extracting *validatory entities* whose presence or absence validates a particular biological interaction. For example, in the following PubMed sentence, "*regulates*" is identified as relational verb relating the biological entities "*Rac1*" and "*transcription of the APP gene*" while "*primary hippocampal neurons*" is identified as *validatory entity*.

''... Racl regulates transcription of the APP gene in primary hippocampal neurons (PMID: l9267423).''

We have also presented a scheme for semantic integration of information extracted from text documents using semantic net. The semantic net highlights the role of a single entity in various contexts, which is useful both for a researcher as well as a layman. The network provides distinct user perspectives and allows navigation over documents with similar information components and is also used to provide a comprehensive view of the collection. It is possible to slice and dice or aggregate to get more detailed or more consolidated view as desired.

The system is also integrated with a biomedical named entity recognizer, ABNER [13], to identify a subset of GENIA ontology concepts (*DNA*, *RNA*, *protein*, *cell-line*, and *cell type*) and tag them accordingly. This helps in answering biological queries formulated at different levels of specificity. Given a query, BioKEVis aims at retrieving all relevant sentences that contain a set of biological concepts stated in a query, in the same context as specified in the query, from the curated database. We have also proposed a document ranking mechanism to present retrieved documents in order of their relevance to user query. The efficacy of BioKEVis is established through experiments on GENIA corpus [28].

The remaining paper is structured as follows: Section 2 presents a review of related works on biomedical text mining. The architectural detail of BioKEVis is discussed in Section 3. Section 4 presents the experimental detail and evaluation of various modules. Section 5 presents a critical discussion to highlight the novelties of the proposed system over existing ones. Finally, Section 6 concludes the paper and provides direction for possible enhancements to the proposed system.

2. Related works

In this section, we present an overview of some of the recent research efforts that have been directed towards the problems of biological relation extraction from text documents. A brief review of the existing biomedical knowledge visualization and query answering systems will be also a part of this section.

2.1. Biological relation extraction

Though, named-entity recognition from biological text documents has gained reasonable success, reasoning about contents of a text document however needs more than identification of the entities present in it. Context of the entities in a document can be inferred from an analysis of the inter-entity relations present in the document. Hence, it is important that the relationships among the biological entities present in a text are also extracted and interpreted correctly. Related works in biological relation extraction can be classified into the following three categories:

Co-occurrence based approach: In this approach, relations between biological entities are inferred based on the assumption that two entities in the same sentence or abstract are related. Negation in the text is not taken into account. Jenssen et al. [21] collected a set of almost 14,000 gene names from publicly available databases and used them to search MEDLINE abstracts. Two genes were assumed to be linked if they appeared in the same abstract; the relation received a higher weight if the gene pair appeared in multiple abstracts. For the pairs with high weights, i.e. with five or more occurrences of the pair, it was reported that 71% of the gene pairs were indeed related. However, the primary focus of the work is to extract related gene pairs rather than studying the nature of these relations. In [32], an ontology-based biological information extraction and query answering (BIEQA) System is proposed which extracts biological relations from MEDLINE abstracts using NLP techniques and co-occurrence based analysis from tagged documents. Each mined relation is associated to a fuzzy membership value, which is proportional to its frequency of occurrence in the corpus and is termed a fuzzy biological relation. The fuzzy biological relations along with other relevant information components like biological entities occurring within a relation, are stored in a database which is integrated with a query-processing module. The query processing module has an interface, which guides users to formulate biological queries at different levels of specificity. The recall values ranged from 84.68% to 86.23% and precision from 94.73% to 98.87%.

Linguistics-based approach: In this approach, usually shallow parsing techniques are employed to locate a set of handpicked verbs or nouns. Rules are specifically developed to extract the surrounding words of these predefined terms and to format them as relations. As with the co-occurrence based approach, negation in sentences is usually ignored. Sekimizu et al. [9] collected the most frequently occurring verbs in a collection of abstracts and developed partial and shallow parsing techniques to find the verb's subject and object. The estimated precision of inferring relations is about 71%. Thomas et al. [10] modified a pre-existing parser based on cascaded finite state machines to fill templates with

information on protein interactions for three verbs – *interact with, associate with, bind to*. They calculated recall and precision in four different manners for three samples of abstracts. The recall values ranged from 24% to 63% and precision from 60% to 81%. The PASTA system is a more comprehensive system that extracts relations between proteins, species and residues [22]. Text documents are mined to instantiate templates representing relations among these three types of elements. This work reports precision of 82% and a recall value of 84% for recognition and classification of the terms, and 68% recall and 65% precision for completion of templates. Ono et al. [11] reports a method for extraction of *protein–protein interactions* based on a combination of syntactic patterns. They employ a dictionary look-up approach to identify proteins in the document. Sentences that contain at least two proteins are selected and parsed with parts-of-speech matching rules. The rules are triggered by a set of keywords, which are frequently used to name protein interactions (e.g., *associate, bind, etc.*). Rinaldi et al. [12] have proposed an approach towards automatic extraction of a pre-defined set of seven relations in the domain of Molecular Biology, based on a complete syntactic analysis of an existing corpus. They extract relevant relations from a domain corpus based on full parsing of the documents and a set of rules that map syntactic structures into the relevant relations. Friedman et al. [23] have developed a natural-language processing system, GENIES, for the extraction of molecular pathways from journal articles. GENIES identifies a predefined set of verbs using templates for each one of these, which are encoded as a set of rules. This work [23] reports a precision of 96% for identifying relations between biological molecules from full-text articles. In [33], the authors have proposed *RelEx* to extract relations between genes and proteins. For relation extraction, the text documents are first converted into dependency parse tree using Stanford Lexicalized Parser. Thereafter, rules are applied to identify candidate relations from parse trees. Both, precision and recall of the proposed system calculated over 1 million MEDLINE abstracts are reported as 80%.

Mixed approach: Ciaramita et al. [18] report an unsupervised learning mechanism for extracting semantic relations between molecular biology concepts from tagged MEDLINE abstracts. For each sentence containing two biological entities, a dependency graph highlighting the dependency between the entities is generated based on linguistic analysis. A relation between two entities is extracted as the shortest path between the pair following the dependency relations. The major emphasis of this work is to determine the role of a concept in a significant relation and enhance the biological ontology to include these roles and relations. Sentences containing complex embedded conjunctions/disjunctions or more than 100 words were not used for relation extraction. In the presence of nested tags, the system considers only the innermost tag.

In [34], Li et al. have developed a framework of kernel-based learning to automatically extract biomedical relations from text documents. They have proposed a novel trace-tree kernel that extends a standard tree kernel by adding a trace kernel to capture richer contextual information. The reported precision and recall values are 70.11% and 64.68%, respectively.

It can be observed that most of the systems have been developed to extract a prespecified set of relations. The relation set is manually chosen to include a set of frequently occurring relations. Each system is tuned to work with a pre-determined set of relations and does not address the problem of relation extraction in a generic way. For example the method of identification of interaction between genes and gene products cannot work for extraction of enzyme interactions from journal articles, or for automatic extraction of protein interactions from scientific abstracts. In line with [18,32], `BioKEVis` attempts to extract generic biological relations along with the associated entities and store them in a structured repository. While mining biological relations the associated prepositions are also considered which very often changes the nature of the verb. Unlike most of the systems mentioned above, `BioKEVis` also identifies the negations in sentences and store them along with the relational verbs. Besides mining relational verbs and the associated entities, the *validatory entities* whose presence or absence validates a particular biological interaction are also identified and stored in the knowledge repository.

2.2. Biomedical knowledge visualization

Though biological relation mining has gained attention of researchers for unraveling the mysteries of biological reactions, their use in biological information visualization is still limited [18]. The powerful combination of precise analysis of the biomedical documents with a set of visualization tools enables the user to navigate and use easily the abundance of biomedical document collection. Visualization is a key element for effective consumption of information. Semantic nets provide a consolidated view of domain concepts and semantic relations among them and can aid in this process. In the information visualization literature, a number of exploratory visualization tools are described in [25]. Zheng et al. [24] have proposed an ontology-based visualization framework, GOClonto, for conceptualization of biomedical document collections. Based on Gene Ontology (GO), GOClonto extracts gene-related terms from biomedical text, applies latent semantic analysis to identify key gene-related terms, allocates documents based on the key gene-related terms, and utilizes GO to automatically generate a corpus-related gene ontology. In [16] a soft-computing based technique is proposed to integrate information mined from biological text documents with the help of biological databases. Castro et al. [17] propose building a semantic net for visualization of relevant information with respect to usecases like the nutrigenomics usecase, wherein the relevant entities around which the semantic net is built are pre-defined.

Although, some visualization methods extract key concepts from document corpora, most of them do not explicitly exploit the semantic relationships between these concepts. The proposed method differs from all these approaches predominantly in its use of pure linguistic techniques rather than use of any pre-existing collection of entities and relations. Moreover, the knowledge visualizer module is integrated with the underlying corpus for comprehending the conceptual structure of biomedical document collections and avoiding information overload for users. On selecting a particular entity or relation in the graph the relevant documents are displayed with highlighting the snippet in which the target knowledge is embedded.

2.3. Biomedical query answering

In order to provide intelligent search mechanisms for extracting relevant information components from a vast collection of text documents a number of biomedical query answering systems appear in the literature. Textpresso [2] is a biological information retrieval and extraction system which analyzes tagged biological documents. Two types of tags are used for tagging text elements manually. The first set of tags defines a collection of biological concepts and the second set of tags defines a set of relations that can relate two categories of biological concepts. A tag is defined by a collection of terms including *nouns*, *verbs*, *etc*. that can be commonly associated to the concept. Portions of the document containing a relevant subset of terms are marked by the corresponding biological concept or relation tag. The search engine allows the user to search for combinations of *concepts*, *keywords* and *relations*. With specific relations like commonly occurring gene–gene interactions, etc. encoded as a relation tag, Textpresso assists the user to formulate semantic queries. The recall value of

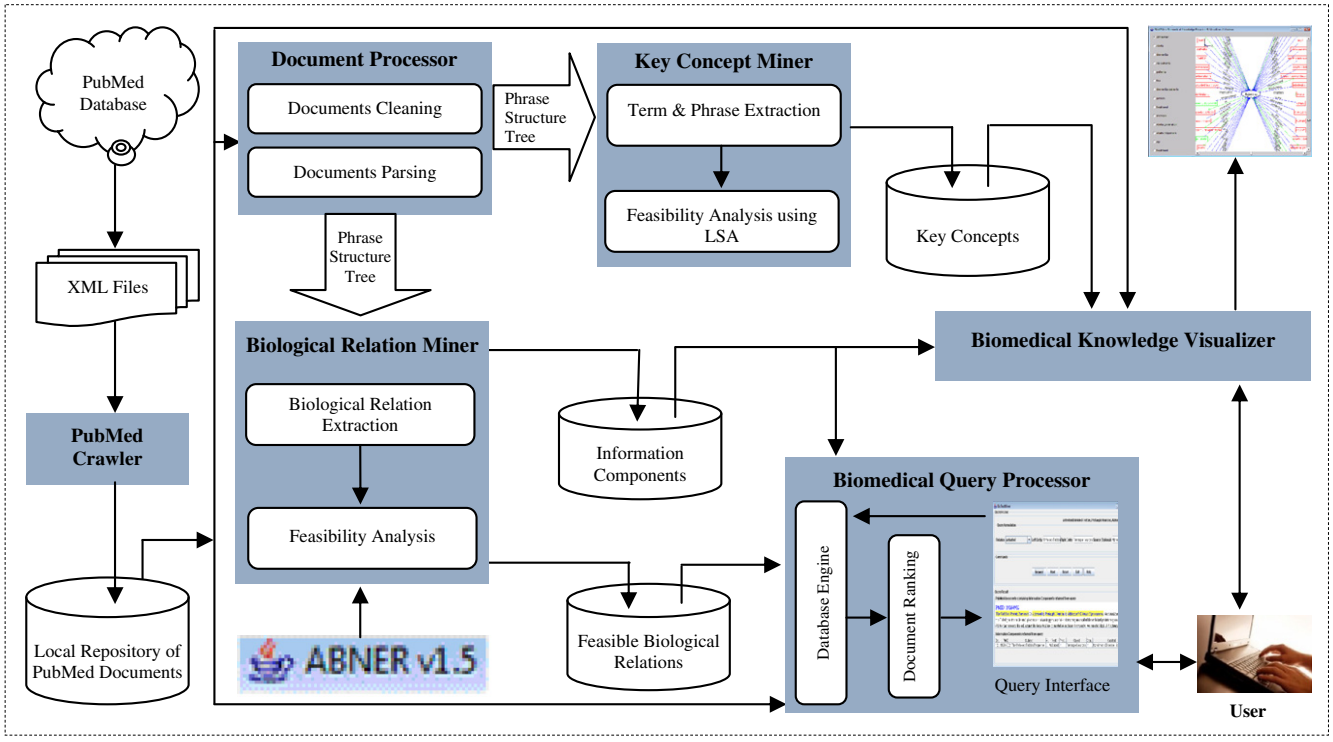

Fig. 1. BioKEVis architecture.

the system is reported to vary from 45% to 95%, depending on whether the search is conducted over abstracts or full text documents. Uramoto et al. [1] have proposed a text-mining system, MedTAKMI, for knowledge discovery from biomedical documents. The system dynamically and interactively mines a large collection of documents with biomedically motivated categories to obtain characteristic information from them. The MedTAKMI system performs entity extraction using dictionary lookup from a collection of two million biomedical entities, which are then used along with their associated category names to search for documents that contain keywords belonging to specific categories. Users can submit a query and receive a document collection in which each document contains the query keywords or their synonyms. The system also uses syntactic information with a shallow parser to extract binary (a verb and a noun) and ternary (two nouns and a verb) relations that are used as keywords by various MedTAKMI mining functions like dictionary-based full text searching, hierarchical category viewer, chronological viewer, etc.

It can be observed that these systems rely on either manual identification of entities and relations or dictionary lookup. In addition, these systems do not use any ranking mechanism to present retrieved documents in order to their relevance of user queries.

3. BioKEVis architecture

In this section, we present the complete architectural detail of BioKEVis which consists of following modules: *PubMed Crawler*, *Document Processor*, *Key Concept Miner*, *Biological Relation Miner*, *Biomedical Knowledge Visualizer*, and *Biomedical Query Processor* (see Fig. 1). The design and working principles of these modules are presented in the following sub-sections.

3.1. PubMed crawler

PubMed Crawler is developed as an interactive module using Java programming language that uses PubMed API (Application Program Interface) to fetch PubMed documents in XML format and store them after parsing into structured database on local machine. Biomedical documents stored in PubMed database are available in XML format in which tags are defined using Document Type Definition[1] (DTD) file standardized by World Wide Web Consortium (W3C). The crawler uses DTD file definitions to create database schema to store fetched XML files from PubMed database into structured form. The fetched XML documents are parsed by crawler to identify different constituents like *PMID*, *title*, *abstract, etc.* to store them in structured database on local machine. There are two types of APIs for parsing XML files – *tree-based Document Object Model (DOM)*, and *event-based Simple API to XML (SAX)*. Our crawler uses the SAX parser as DOM parser requires to read in and store the entire document in main memory prior to writing out any data and it is not possible for a large file that do not fit in the memory. However, the SAX parser receives data through a stream and recognizes the beginning and end of a document, element, or attribute in an event driven manner. It writes out the data as it proceeds and there is no need to load entire file in the memory. After parsing XML files the JDBC is used to store parsed data into the database.

3.2. Document processor

The *Document Processor* fetches the text documents from local database repository for parts-of-speech (POS) analysis which assigns POS tags to every word in a sentence, where a tag reflects the syntactic category of the word [4]. The POS tags are useful to identify the grammatical structure of sentences like noun and verb phrases and their inter-relationships. For POS analysis we have used the Stanford parser,[2] which is a statistical parser. The Stanford parser receives documents as input and works out the grammatical structure of sentences to convert them into equivalent phrase structure tree. A list of sample sentences and their corresponding

[1] http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd.
[2] http://nlp.stanford.edu/downloads/lex-parser.shtml.

Table 1
A List of PubMed abstracts and corresponding phrase structure tree generated through Stanford Parser.

PMID	Sentence	Phrase structure tree
19295912	Transcriptome analysis of synaptoneurosomes identifies neuroplasticity genes overexpressed in incipient Alzheimer's disease	(ROOT (S (NP (NP (JJ Transcriptome) (NN analysis)) (PP (IN of) (NP (NNS synaptoneurosomes)))) (VP (VBZ identifies) (NP (NP (JJ neuroplasticity) (NNS genes)) (VP (VBN overexpressed) (PP (IN in) (NP (NP (JJ incipient) (NNP Alzheimer) (POS 's)) (NN disease)))))) (. .)))
19295164	Recent studies suggest that bone marrow-derived macrophages can effectively reduce beta-amyloid (Abeta) deposition in brain	(ROOT (S (NP (JJ Recent) (NNS studies)) (VP (VBP suggest) (SBAR (IN that) (S (NP (JJ bone) (JJ marrow-derived) (NNS macrophages)) (VP (MD can) (ADVP (RB effectively)) (VP (VB reduce) (NP (NP (JJ beta-amyloid) (PRN (-LRB- -LRB-) (NP (NNP Abeta)) (-RRB- -RRB-)) (NN deposition)) (PP (IN in) (NP (NN brain)))))))))) (. .)))
19275635	There is substantial and compelling evidence that aggregation and accumulation of amyloid beta protein (Abeta) plays a pivotal role in the development of Alzheimer's disease (AD)	(ROOT (S (S (NP (EX There)) (VP (VBZ is) (NP (ADJP (JJ substantial) (CC and) (JJ compelling)) (NN evidence)) (SBAR (IN that) (S (NP (NP (NN aggregation) (CC and) (NN accumulation)) (PP (IN of) (NP (NP (JJ amyloid) (JJ beta) (NN protein)) (PRN (-LRB- -LRB-) (NP (NNP Abeta)) (-RRB- -RRB-))))) (VP (VBZ plays) (NP (NP (DT a) (JJ pivotal) (NN role)) (PP (IN in) (NP (NP (DT the) (NN development)) (PP (IN of) (NP (NP (NNP Alzheimer) (POS 's)) (NN disease)) (PRN (-LRB- -LRB-) (NNP AD) (-RRB- -RRB-)))))))))) (: ;)
19263040	Memory deficits and neurochemical changes induced by C-reactive protein in rats: implication in Alzheimer's disease	(ROOT (NP (NP (NP (NN Memory) (NNS deficits) (CC and) (NN neurochemical) (NNS changes)) (VP (VBN induced) (PP (IN by) (NP (NP (JJ C-reactive) (NN protein)) (PP (IN in) (NP (NNS rats)))))))) (: :) (NP (NP (NN implication)) (PP (IN in) (NP (NP (NNP Alzheimer) (POS 's)) (NN disease)))) (. .)))

phrase structure tree generated by Stanford parser is shown in Table 1. These sentences are also referred in rest of the paper to explain the functioning details of other modules.

3.3. Key concept miner

The phrase structure tree generated by document processor is further analyzed by this module to identify *feasible key concepts* for conceptualization of document corpus. The key steps in this process are: *term and phrase extraction*, and *feasibility analysis using LSA*. These steps are explained in the following sub-sections.

3.3.1. Term and phrase extraction

For term and phrase extraction, we consider only those internal NP (noun phrase) nodes whose child nodes appear as leaf in phrase structure tree. If a node NP has single child node tagged as *noun* it is extracted as *term*. If NP has two or more child nodes then the string concatenation function is applied to club the child nodes, tagged as *noun* or *adjective*, together and it is identified as *phrase*. Hence, a term is a noun phrase containing single word. The lists of terms and phrases are compiled separately for the purpose of feasibility analysis using LSA as discussed in the following section [27]. After compiling the lists, the terms having a match in the list of stop-words[3] are filtered out and phrases starting or ending with stop-words are cleaned after removing the stop-words from them. In addition, the terms containing only numeric and special characters or having length (number of characters) less than three are also removed from the lists. For remaining phrases we calculate their weight using term frequency (*tf*) and inverse document frequency (*idf*) in each document of the corpus [29]. The weight of a phrase p_i in jth document, $\omega(p_{i,j})$, is calculated using Eqs. (1) and (2) where, $\mathrm{tf}(p_{i,j})$ is the number of times p_i occurs in jth document. $|D|$ is the total number of documents in the corpus, and $|\{d_j : p_i \in d_j\}|$ is the number of documents where p_i appears. While counting frequency of a term or phrase they are stemmed using Porter's stemmer [30]. All those phrases having normalized average weight over all documents above a threshold are retained for feasibility analysis using LSA.

The `TermPhraseExtraction` algorithm given in Table 2 presents the process of term and phrase extraction and weight-matrix generation in a formal way. A partial list of identified terms and phrases from text documents on *Alzheimer disease* are shown in Tables 3 and 4, respectively. The terms and phrases are ranked in non-increasing order of their normalized weights

$$\omega(p_{i,j}) = tf(p_{i,j}) \times idf(p_i) \tag{1}$$

$$idf(p_i) = \log\left(\frac{|D|}{|\{d_j : p_i \in d_j\}|}\right) \tag{2}$$

Table 2
Algorithm for term and phrase extraction, and their weight calculation.

Algorithm: TermPhraseExtraction(F)

Input: A forest F of phrase structure trees
Output: List of terms L_{Term}, their idf vector and weight matrix; list of phrases L_{Phrase}

```
L_Term ← φ, L_Phrase ← φ
For each T ∈ F do // consider each phrase structure tree
  For each internal NP node λ ∈ T do // consider each noun phrase node
    If all child nodes of λ are leaf node then
      p = "" // Initialize phrase as null string
      For each node ξ ∈ child[λ] do
        If (tag(ξ) = NN* OR tag(ξ) = JJ) then
          p ← p + word(ξ)
          If (tag(ξ) = NN*) then
            L_Term ← L_Term ∪ word(ξ)
          End if
        End if
      End for
      L_Phrase ← L_Phrase ∪ p
    End if
  End for
End for
For i ← 1 to length(L_Term) do
  For j ← 1 to n do // n is total number of documents
    W[i][j] ← tf(t_i,j) × idf(t_i)
  End for
  If AvgWeight(t_i) < θ_1 then // θ_1 is a threshold value
    L_Term ← L_Term − t_i
  End if
End for
For each p_i ∈ L_Phrase do
  If AvgWeight(np_i) < θ_2 then // using Eq. (1)
    L_Phrase ← L_Phrase − np_i
  End if
End for
```

Table 3
A partial list of terms and their normalized weights extracted from a corpus containing PubMed Abstracts on "Alzheimer disease".

Term (t)	$\omega(t)$	Term (t)	$\omega(t)$	Term (t)	$\omega(t)$
AD	1.00	Protein	0.45	Mice	0.32
Abeta	0.86	APP	0.43	Expression	0.32
Patients	0.66	MCI	0.38	Risk	0.31
Dementia	0.62	Impairment	0.35	Memory	0.31
Disease	0.59	Levels	0.35	Results	0.30
Brain	0.51	Study	0.34	Neurons	0.30
Alzheimer	0.49	Treatment	0.33	Studies	0.29
Tau	0.48	Cells	0.33	Activity	0.29

[3] A list of 500 stop-words appears at http://www.abulaish.com/stopwords.txt.

3.3.2. Feasibility analysis using LSA

Latent Semantic Analysis (LSA) is a technique which is used to analyze relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms [26]. This is applied to further boost the precision of key concept extraction process, discussed in the previous section. For LSA each document d is represented as a feature vector $\vec{d} = (w_{t_1}, \ldots, w_{t_m})$, where m is the number of terms, and w_{t_i} is the weight of term t_i in document d as calculated in the previous section. Feature vector for each document in the corpus is used to gen-

(a)

D1: Membrane cholesterol enrichment prevents Abeta-induced oxidative stress in Alzheimer's fibroblasts.
D2: Dementia is a chronic progressive mental disorder, which adversely affects memory, thinking, comprehension, calculation and language.
D3: Transcriptome analysis of synaptoneurosomes identifies neuroplasticity genes overexpressed in incipient Alzheimer's disease.
D4: Alzheimer's disease is associated with an increased risk of unprovoked seizures.
D5: Early cognitive deficit characteristic of early Alzheimer's disease seems to be produced by the soluble forms of beta-amyloid protein.
D6: Alzheimer's disease (AD) and stroke are two leading causes of age-associated dementia.
D7: Alzheimer's disease (AD) is associated with intact experience but abnormal expression of emotion.
D8: Aggregated fibrillary microtubule-associated protein tau is the major component of neurofibrillary tangles in Alzheimer's disease.

(b)

T1: disease
T2: Alzheimer
T3: Dementia
T4: Protein
T5: AD
T6: Cholesterol
T7: Enrichment
T8: Fibroblasts
T9: Oxidative

(c)

P1: Cholesterol enrichment
P2: Oxidative stress

Fig. 2. Sample text documents along with terms and phrases present therein to illustrate LSA process.

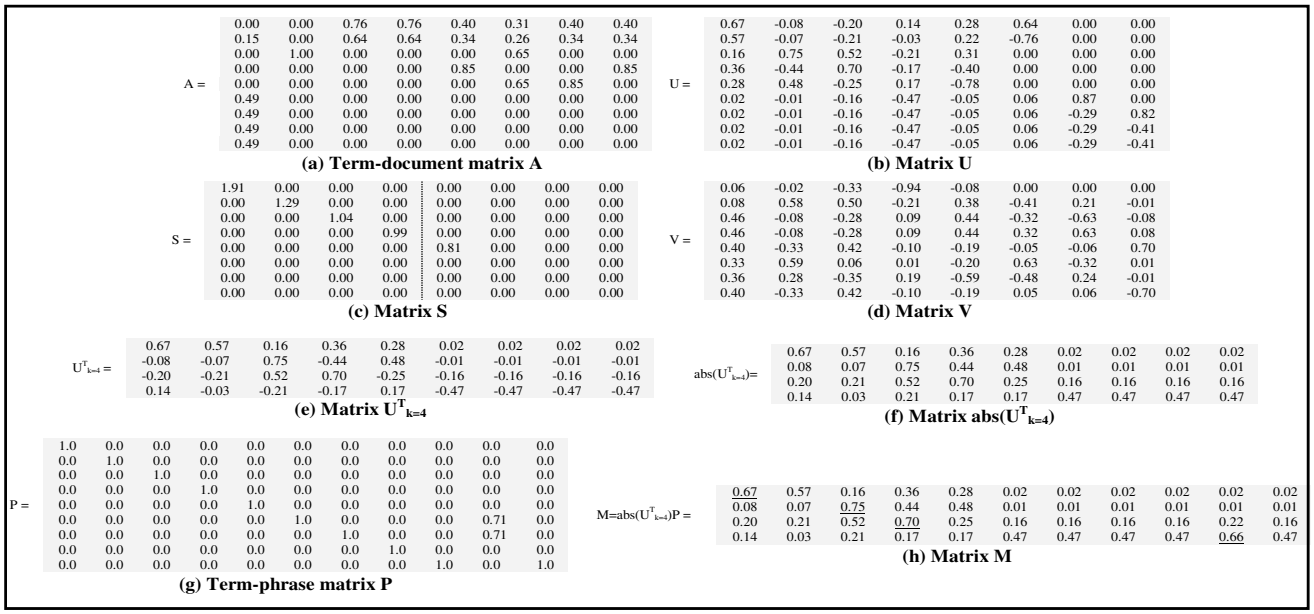

Fig. 3. Matrices used during LSA for key concept identification.

Table 4
A partial list of phrases and their normalized weights extracted from a corpus containing PubMed Abstracts on "Alzheimer disease".

Phrase (p)	$\omega(p)$	Phrase (p)	$\omega(p)$	Phrase (p)	$\omega(p)$
Alzheimer disease	1.00	Mouse model	0.34	Music therapy	0.26
AD patients	0.88	Tau protein	0.33	Weight loss	0.26
Cognitive impairment	0.83	Tau phosphorylation	0.29	Disease progression	0.25
Precursor protein	0.75	Brain injury	0.29	Control subjects	0.25
Risk factors	0.56	Gene expression	0.29	Abeta aggregation	0.24
Vascular dementia	0.52	Oxygen species	0.29	Risk factor	0.24
Cell death	0.48	AD pathogenesis	0.27	Abeta peptides	0.24
Control group	0.34	Resonance imaging	0.26	Dementia patients	0.23

erate *term-document* matrix by composing feature vectors of all the documents in the corpus. In this matrix, a column vector represents a document and a row vector represents a term as document's feature. For example, in term-document matrix A shown in Fig. 3(a), the rows represent the terms listed in Fig. 2(b), i.e., first row represents the term "*disease*", second row represents "*Alzheimer*" and so on. Similarly, the columns in Fig. 3(a) represent the documents listed in Fig. 2(a), i.e., the first column corresponds to document D_1, the second column to D_2, and so on. In matrix A all column vectors are normalized so that their length is 1. Thereafter, Singular Value Decomposition (SVD) is applied on A which breaks it into three matrices U, S, and V, shown in Fig. 3(b), (c), and (d), respectively, such that $A = USV^T$. SVD translates the term and document vectors into a concept space. The first r columns of U (where r is A's rank) form an orthogonal basis for the matrix A's term space. Therefore, basis vectors, which are column vectors in U, represent abstract terms of corresponding document. In practice, it is not possible to take all r abstract terms. Therefore we take a threshold value, θ, and find the number of singular values (say k) in matrix S that is higher than this θ. Then, we use U_k, which consists of first k columns of U as shown in Fig. 3(e), to obtain k most important terms for the document corpus. At the time of identification of important terms and phrases we consider only magnitude therefore we take absolute value of U_k as shown in Fig. 3(f)

$$(p_{i,j}) = \begin{cases} idf(t_i), & \text{if } i = j \text{ and } j \leqslant m \\ idf(t_i), & \text{if } j > m \text{ and } t_i \text{ is a substring of the noun phrase} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Since the column vectors in U represent the importance of the terms for the document corpus, we also use U to evaluate the importance of phrases. For this, we construct a matrix P of order $m \times (m + p)$, where m and p represent the number of terms and phrases respectively. In matrix P, each row represents a term and columns represent terms as well phrases. Elements of matrix P are computed using Eq. (3). Like term-document matrix A, the column vector lengths in P are also normalized to 1 as shown in Fig. 3(g).

Table 5
Algorithm to identify feasible key concepts

Algorithm: FeasibilityAnalysis(A, L_{Term}, L_{Phrase}, Term_idf)

Input: Term-document weight matrix (A), list of terms(L_{Term}), list of phrases(L_{Phrase}), and Term_idf
Output: A list $L_{KeyConcepts}$ of key concepts
1. [U,S,V] ← SVD(A,0)// Decompose matrix A into U, S, V matrices so that A = USV^T
// Construct matrix P using Eq. (3) with the help of L_{Term}, L_{Phrase} and Term_idf
2. M ← Abs(U_k^T) × P // where k < r, the rank of A
3. $L_{KeyConcepts}$ ← φ
4. For i = 1 to rows(M) do
5. max ← M(i, 1)
6. For j = 2 to cols(M) do
7. If (M(i,j) > max) then
8. max ← M(i,j)
9. End if
10. End for
11. For j = 1 to cols(M) do
12. If (M(i,j) = max) then
13. If (j ⩽ m) then // m = length(L_{Term})
14. $L_{KeyConcepts}$ ← $L_{KeyConcepts}$ ∪ {L_{Term}[j]}
15. Else
16. $L_{KeyConcepts}$ ← $L_{KeyConcepts}$ ∪ {L_{Phrase}[j-m]}
17. End if
18. End if
19. End for
20. End for
21. Return $L_{KeyConcepts}$

Thereafter, the matrix $abs(U_k^T)$ is multiplied with P to get matrix M as shown in Fig. 3(h) which represents the importance of terms and phrases. In matrix M, the highest value in each row is identified and the corresponding term or phrase is extracted as feasible key concept. In Fig. 3(h), the highest value in each row is underlined and the corresponding terms and phrases identified as key concepts are: *disease*, *dementia*, *protein*, and *cholesterol enrichment*. The algorithm `FeasibilityAnalysis` given in Table 5 presents the feasibility analysis process formally. A partial list of feasible key concepts extracted from a collection of PubMed abstracts on *Alzheimer disease* is shown in Table 6. The performance of LSA over *tf-idf*, evaluated on GENIA corpus [28], to identify key concepts is shown in Fig. 4. One of the major difficulties in terms of memory space while using SVD for latent semantic analysis of unstructured texts is to handle high-order sparse term-document matrix. To overcome this problem the sparse matrix methods for SVD [31] can be used.

3.4. Biological relation miner

A biological relation is assumed to be binary in nature, which defines a specific association between an ordered pair of biological entities. The process of identifying biological relations is accomplished in two stages. During the first stage, prospective information components (Definition 1) which might embed biological relations within them are identified from the sentences. During the second stage, a feasibility analysis is employed to identify correct biological relations. These steps are explained in the following sub-sections.

Definition 1 (*Information Component*). An Information Component (IC) is a 7-tuple of the form $\langle E_i, A, V, P_v, E_j, P_c, E_k \rangle$ where, E_i, and E_j are noun phrases associated by V which is a relational verb; A is adverb; P_v is verbal-preposition associated with V; E_k is validatory phrase associated with E_j through conjunctional-preposition P_c.

Table 6
A partial list of feasible key concepts extracted from a corpus containing PubMed Abstracts on "*ALZHEIMER DISEASE*".

Key concept (terms)		Key concept (phrases)	
AD	APP	AD patients	Care physicians
Abeta	Treatment	Gene expression	Clinical Trials
Patients	Mice	Music therapy	Abeta generation
Dementia	Expression	Weight loss	Side effects
Tau	Memory	Dementia patients	Abeta oligomers
Protein	Neurons	Neurodegenerative disorders	

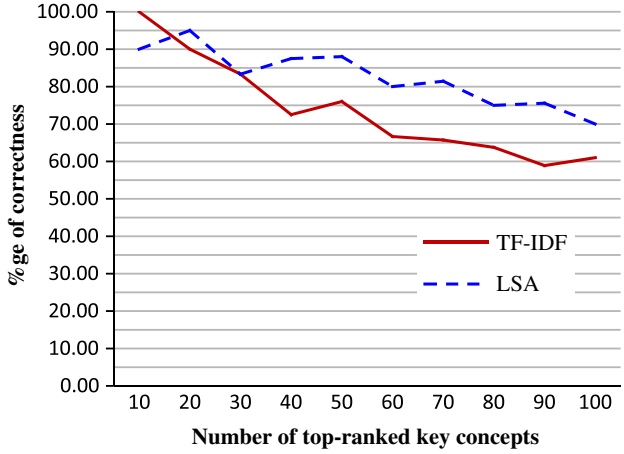

Fig. 4. Performance comparison of *LSA* and *tf-idf* to identify key concepts.

Table 7
Algorithm to extract information components from phrase structure tree.

Algorithm: InformationComponentExtraction(T)

Input: Phrase structure tree T, created though Stanford parser
Output: A list of Information Components L_{IC}

1. $L_{IC} \leftarrow \varphi$
2. For each node N ∈ T do
3. IC ← φ
4. For each child $\eta_i \in N$ do
5. If η_i = NP AND η_j = VP AND i < j then
6. If $\lambda_0 \in child[\eta_j]$ = V AND $\lambda_i \in child[\eta_j]$ = NP AND $\lambda_j \in child[\eta_j]$ = PP AND i ≠ 0, j ≠ 0, i < j AND $\xi_0 \in child[\lambda_j]$ = p AND $\xi_i \in child[\lambda_j]$ = NP AND i ≠ 0 then
7. IC = ⟨E(η_i), null, V, null, E(λ_i), p, E(ξ_i)⟩ //E(x) represents the entity extracted from the subtree rooted at x
8. Else if $\lambda_0 \in child[\eta_j]$ = V AND $\lambda_i \in child[\eta_j]$ = NP AND i ≠ 0 then
9. If $\xi_i \in child[\lambda_i]$ = NP AND $\xi_j \in child[\lambda_i]$ = PP AND i < j AND $\tau_0 \in child[\xi_j]$ = p AND$\tau_i \in child[\xi_j]$ = NP AND i ≠ 0 then
10. IC = ⟨E(η_i), null, V, null, E(ξ_i), p, E(τ_i)⟩
11. Else IC = ⟨E(η_i), null, V, null, E(λ_i), null, null⟩
12. End if
13. Else if $\eta_k \in child[\eta_j]$ = VP AND $\lambda_0 \in child[\eta_k]$ = V AND $\lambda_i \in child[\eta_k]$ = NP AND $\lambda_j \in child[\eta_k]$ = PP AND i ≠ 0, j ≠ 0, i < j AND $\xi_0 \in child[\lambda_j]$ = p AND $\xi_i \in$ - $child[\lambda_j]$=NP AND i ≠ 0 then
14. IC = ⟨E(η_i), null, V, null, E(λ_i), p, E(ξ_i)⟩
15. Else if $\eta_k \in child[\eta_j]$ = VP AND $\lambda_0 \in child[\eta_k]$ = V AND $\lambda_i \in child[\eta_k]$ = NP AND i ≠ 0 then
16. If $\xi_i \in child[\lambda_i]$ = NP AND $\xi_j \in child[\lambda_i]$ = PP AND i < j AND $\tau_0 \in child[\xi_j]$ = p AND $\tau_i \in child[\xi_j]$ = NP AND i ≠ 0 then
17. IC = ⟨E(η_i), null, V, null, E(ξ_i), p, E(τ_i)⟩
18. Else IC = ⟨E(η_i), null, V, null, E(λ_i), null, null⟩
19. End if
20. Else if $\lambda_0 \in child[\eta_j]$ = V AND $\lambda_i \in child[\eta_j]$ = PP AND $\lambda_j \in child[\eta_j]$ = PP AND i ≠ 0, j ≠ 0, i < j AND £$_0 \in child[\lambda_i]$ = p_1 AND £$_m \in child[\lambda_i]$ = NP AND m ≠ 0 AND $\xi_0 \in child[\lambda_j]$ = p_2 AND $\xi_i \in child[\lambda_j]$ = NP AND i ≠ 0 then
21. IC = ⟨E(η_i), null, V, p_1, E(£$_m$), p,E(ξ_i)⟩
22. Else if $\lambda_0 \in child[\eta_j]$ = V AND $\lambda_i \in child[\eta_j]$ = PP AND £$_0 \in child[\lambda_i]$ = p_1 AND £$_m \in child[\lambda_i]$ = NP AND m ≠ 0 then
23. If $\xi_i \in child[£_m]$ = NP AND $\xi_j \in child[£_m]$ = PP AND i < j AND $\tau_0 \in child[\xi_j]$ = p_2 AND $\tau_i \in child[\xi_j]$ = NP AND i ≠ 0 then
24. IC = ⟨E(η_i), null, V, p_1, E(ξ_i), p_2, E(τ_i)⟩
25. Else IC = ⟨E(η_i), null, V, p_1, E(£$_m$), null, null⟩
26. End if
27. Else if $\eta_k \in child[\eta_j]$ = VP AND $\lambda_0 \in child[\eta_k]$ = V AND $\lambda_i \in child[\eta_k]$ = PP AND $\lambda_j \in child[\eta_k]$ = PP AND i ≠ 0, j ≠ 0, i < j AND £$_0 \in child[\lambda_i]$ = p_1 AND £$_m \in child[\lambda_i]$ = NP AND m ≠ 0 AND $\xi_0 \in child[\lambda_j]$ = p_2 AND $\xi_i \in child[\lambda_j]$ = NP AND i ≠ 0 then
28. IC = ⟨E(η_i), null, V, p_1, E(£$_m$), p_2, E(ξ_i)⟩
29. Else if $\eta_k \in child[\eta_j]$ = VP AND $\lambda_0 \in child[\eta_k]$ = V AND $\lambda_i \in child[\eta_k]$ = PP AND i ≠ 0 AND £$_0 \in child[\lambda_i]$ = p_1 AND £$_m \in child[\lambda_i]$ = NP AND m ≠ 0 then
30. If $\xi_i \in child[£_m]$ = NP AND $\xi_j \in child[£_m]$ = PP AND i < j AND $\tau_0 \in child[\xi_j]$ = p_2 AND $\tau_i \in child[\xi_j]$ = NP AND i ≠ 0 then
31. IC = ⟨E(η_i), null, V, p_1, E(ξ_i), p_2, E(τ_i)⟩
32. Else IC = ⟨E(η_i), null, V, null, E(£$_m$), null, null⟩
33. End if
34. Else if $\eta_k \in child[\eta_j]$ = VP AND $\eta_l \in child[\eta_k]$ = VP AND $\lambda_0 \in child[\eta_l]$ = V AND $\lambda_i \in child[\eta_l]$ = NP AND $\lambda_j \in child[\eta_l]$ = PP AND i ≠ 0, j ≠ 0, i < j AND £$_0 \in child[\lambda_i]$ = p_1 AND $\xi_0 \in child[\lambda_j]$ = p AND $\xi_i \in child[\lambda_j]$ = NP AND i ≠ 0 then
35. IC = ⟨E(η_i), null, V, null, E(λ_i), p, E(ξ_i)⟩
36. Else if $\eta_k \in child[\eta_j]$ = VP AND $\eta_l \in child[\eta_k]$ = VP AND $\lambda_0 \in child[\eta_l]$ = V AND $\lambda_i \in child[\eta_l]$ = NP AND i ≠ 0 then
37. If $\xi_i \in child[\lambda_i]$ = NP AND $\xi_j \in child[\lambda_i]$ = PP AND i < j AND $\tau_0 \in child[\xi_j]$ = p AND $\tau_i \in child[\xi_j]$ = NP AND i ≠ 0 then
38. IC = ⟨E(η_i), null, V, null, E(ξ_i), p, E(τ_i)⟩
39. Else IC = ⟨E(η_i), null, V, null, E(λ_i), null, null⟩
40. End if
41. Else if $\lambda_0 \in child[\eta_j]$ = V AND $\lambda_i \in child[\eta_j]$ = ADVP AND $\lambda_j \in child[\eta_j]$ = PP AND i ≠ 0, j ≠ 0, i < j AND $\lambda_k \in child[\lambda_i]$ = PP AND £$_0 \in child[\lambda_k]$ = p_1AND £$_m \in child[\lambda_k]$ = NP AND m ≠ 0 AND $\xi_0 \in child[\lambda_j]$ = p_2 AND $\xi_i \in child[\lambda_j]$ = NP AND i ≠ 0 then
42. IC = ⟨E(η_i), adv, V, p_1, E(£$_m$), p, E(ξ_i)⟩
43. Else if $\lambda_0 \in child[\eta_j]$ = V AND $\lambda_i \in child[\eta_j]$ = ADVP AND i ≠ 0 AND $\lambda_k \in child[\lambda_i]$ = PP AND £$_0 \in child[\lambda_k]$ = p_1 AND £$_m \in child[\lambda_k]$ = NP AND m ≠ 0 then
44. If $\xi_i \in child[£_m]$ = NP AND $\xi_j \in child[£_m]$ = PP AND i < j AND $\tau_0 \in child[\xi_j]$ = p_2 AND $\tau_i \in child[\xi_j]$ = NP AND i ≠ 0 then
45. IC = ⟨E(η_i), adv, V, p_1, E(ξ_i), p_2, E(τ_i)⟩
46. Else IC = ⟨E(η_i), adv, V, p_1, E(£$_m$), null, null⟩
47. End if
48. End if
49. End if
50. End for
51. If IC ≠ φ then
52. $L_{IC} \leftarrow L_{IC} \cup$ IC
53. End if
54. End for
55. Return L_{IC}

3.4.1. Information component extraction

An *information component* is usually manifested in a document centered around relational verb. The proposed approach to information component extraction traverses the phrase structure tree and analyzes the phrases and their linguistic dependencies in order to trace relational verbs and other constituents. Since the entities are marked as terms and phrases, this module exploits the phrase boundary and proximitivity, to identify relevant information components. Initially all tuples of the form $\langle E_i, A, V, P_t, E_j, P_c, E_k \rangle$ are retrieved from text documents.

Since a verb may occur in a sentence in its root form or as a variant of it different classes of variants of a relational verb are recognized by our system. The first of this class comprises of *morphological variants* of the root verb, which are essentially modifications of the root verb itself. In English language the word *morphology* is usually categorized into "inflectional" and "derivational"

morphology. *Inflectional morphology* studies the transformation of words for which the root form only changes, keeping the syntactic constraints invariable. For example, the root verb "*activate*", has three inflectional verb forms – "*activates*", "*activated*" and "*activating*". *Derivational morphology* on the other hand deals with the transformation of the stem of a word to generate other words that retain the same concept but may have different syntactic roles. Thus, "*activate*" and "*activation*" refer to the concept of "making active", but one is a verb and the other one a noun. Similarly, *inactivate*, *transactivate*, *deactivate*, etc. are derived morphological variants created with addition of prefixes. Our system considers both derivational and inflectional variants of a root verb.

In the context of biological relations, we also observe that the occurrence of a verb in conjunction with a preposition very often changes the nature of the verb. For example, the functions associated to the verb "*activates*" may be quite different from the ones that can be associated to the verb form "*activates in*", in which the verb "*activates*" is followed by the preposition "*in*". Thus our system also considers biological relations represented by a combination of *root verbs or their morphological variants, and prepositions* that follow these. Typical examples of biological relations identified in this category include "*activated in*", "*binds to*", "*stimulated with*", *etc.*, which denotes a significant class of biological reactions. Besides mining relational verbs with accompanying prepositions and associated entities, the entities associated with object entity through conjunctional prepositions are also extracted and termed as *validating entity*, which presence or absence validates a particular biological interaction.

Information component extraction process is implemented as a rule-based system. Dependencies output by the parser are analyzed to identify *subject*, *object*, *verb*, *preposition*, and various other relationships among elements in a sentence. The `InformationComponentExtraction` algorithm shown in Table 7 dictates the implementation detail of the rule-based system. A partial list of information components extracted by this algorithm from PubMed documents (given in Table 1) is shown in Table 8. The biological entities appearing in information components are marked with a biological entity recognizer that helps in identifying valid biological relations and answering user queries based on biological concepts. For this purpose, the `BioKEVis` is integrated with a biological entity recognizer, ABNER v1.5 [13], which is an open source software tool for molecular biology text mining. It is a machine learning system using conditional random fields with a variety of orthographic and contextual features. It also includes a Java application programming interface allowing users to incorporate ABNER into their own systems and train models on new corpora. ABNER is trained for NLPBA corpus to identify five biological entities – *protein*, *DNA*, *RNA*, *cell line*, and *cell type* with average precision and recall values as 69.1% and 72.0%, respectively. It is also trained for BioCreative corpus to identify protein/gene with average precision and recall values as 74.5% and 65.0%, respectively.

3.4.2. Feasible biological relation identification

A biological relation is usually manifested in a document as a relational verb associating two or more biological entities. The biological actors associated to a relation can be inferred from the biological entities located in the proximity of the relational verb. At present, we have considered only binary relations. In order to compile biological relations from information components, we consider only those tuples in which either subject or object field has at least one biological entity. This consideration deals with the cases in which pronouns are used to refer the biological entities appearing in previous sentences. In this way, a large number of irrelevant verbs are eliminated from being considered as biological relations. Since, our aim is not just to identify possible relational verbs but to identify feasible biological relation. Hence, we engage in statistical analysis to identify feasible biological relations. To consolidate the final list of feasible biological relations we take care of two things. Firstly, since various forms of the same verb represent a basic biological relation in different forms, the feasible collection is extracted by considering only the unique root forms after analyzing the complete list of information components. The root verb having frequency count greater than or equal to a threshold value is retained as root biological relations. Thereafter, information components are again analyzed to identify and extract the morphological variants of the retained root verbs.

The core functionalities of the biological relation and morphological variants finding module is summed up in the following steps.

- Let L_V be the collection of verbs or verb–preposition pairs, which are extracted as part of information components. Each verb can occur in more than one form in the list L_V. For example, the verb *activate* may occur in the form of *activate*, *activates*, *activated* or *activated in*, *etc.*, all of them essentially representing the biological interaction "*activation*" in some form. The list L_V is analyzed to determine the set of unique root forms. The frequency of occurrence of each root verb is the sum-total of its occurrence frequencies in each form. All root verbs with frequency less than a user-given threshold are eliminated from further consideration. The surviving verbs are stored in L_{RV} and termed as *most-frequently occurring* root verbs representing important *biological relations*.
- Once the frequent root verb list is determined, a pattern matching technique is applied on L_V to identify and extract the morphological variants of all root verbs in L_{RV}.

Algorithm `BiologicalRelationExtraction` given in Table 9 defines this process formally. A partial list of feasible biological

Table 8
A partial list of information components extracted from the example documents on "Alzheimer disease" given in Table 1.

Left entity	Adverb	Relational verb	Verbal prep.	Right entity	Conjunction preposition	Validatory phrase	PubMed ID
the CST3 gene	not	associated	with	AD risk	in	the Finnish population	19293566
neuroplasticity genes	–	overexpressed	in	incipient Alzheimer's disease	–	–	19295912
global measures of cognition	–	declined	with	increasing levels of dimeric Abeta (dAbeta)	–	–	19295912
bone marrow-derived macrophages	–	reduce	–	beta-amyloid (Abeta) deposition	in	brain	19295164
aggregation and accumulation of amyloid beta protein (Abeta)	–	plays	–	a pivotal role	in	the development of Alzheimer 's disease (AD)	19275635
memory deficits and neurochemical changes	–	induced	by	C-reactive protein	in	rats	19263040

Table 9
Algorithm to extract biological relations.

Algorithm: BiologicalRelationExtraction(L_{IC})

Input: L_{IC} – A list of information components
Output: A set R of feasible biological relations and their morphological variants

1. $L_V \leftarrow \varphi$, $L_{UV} \leftarrow \varphi$, $L_{RV} \leftarrow \varphi$
2. For all IC $\in L_{IC}$ do
3. If $E_i \in$ IC.subject OR $E_i \in$ IC.object then // E_i is a biological entity identified by ABNER
4. $L_V \leftarrow L_V \cup$ IC.verb + IC.preposition
5. End if
6. End for
7. $L_{UV} \leftarrow$ UNIQUE(L_V) // create a list of unique verbs
8. Filter out verbs from L_{UV} with a prefix as ξ, where $\xi \in$ {cross-, extra-, hydro-, micro-, milli-, multi-, photo-, super-, anti-, down-, half-, hypo-, mono-, omni-, over-, poly-, self-, semi-, tele-, dis-, epi-, mis-, non-, pre-, sub-, de-, di-, il-, im-, ir-, un-, up-}
9. Filter out verbs from L_{UV} with a suffix as $\Im$, where $\Im \in$ {-able, -tion, -ness, -less, -ment, -ally, -ity, -ism, -ous, -ing, -er, -or, -al, -ly, -ed, -es, -ts, -gs, -ys, -ds, -ws, -ls, -rs, -ks, -en}
10. For all V $\in L_{UV}$ do
11. N = FreqCount(V)
12. If N $\geqslant \theta$ {threshold value} then
13. $L_{RV} \leftarrow L_{RV} \cup$ V
14. End if
15. End for
16. R $\leftarrow L_{RV}$
17. For all $V_i \in L_{RV}$ do //identifying morphological variants
18. For all $V_j \in L_{UV}$ do
19. If $V_i \in$ SubString(V_j) then
20. R $\leftarrow$ R $\cup V_j$
21. End if
22. End for
23. End for
24. Return R

relations and their morphological variants extracted from a corpus of 500 PubMed abstracts related to *Alzheimer disease* is shown in Table 10.

3.5. Biomedical knowledge visualizer

One of the crucial requirements when developing a text mining system is the ability to browse through the document collection and be able to *visualize* various elements within the collection. This type of interactive exploration enables the identification of new types of entities and relationships that can be extracted for better exploration of results from the information extraction phase [19,20]. *Semantic net* created as relationship maps provides a visual means for concise representation of relationships among key terms in a given context.

Table 10
A partial list of feasible biological relations and their morphological variants extracted from a corpus of 500 PubMed abstracts related to "Alzheimer disease".

Biological relations	Morphological variants
associate	associate with, associated with, associated to
increase	increased, increases, increased in, increased after, increased by, increased over
induce	induced, induced by, induces, induced in, induced with
investigate	investigated, investigated in, investigates, investigated by, investigated with, investigated for
show	showed, shown, shown on, show for, shows
reduce	reduced, reduces, reduced by, reduced in
decreased	decreased in, decreased as, decreased with, decreased across
observed	observed in, observed between, observed for, observed over
use	used, used for, used in
regulate	regulated by, regulates
affect	affected, affects, affected in, affected by, affecting
express	expressed in, expressing, express as, expresses, expressed from
attenuate	attenuated, attenuated by, attenuates, attenuated in
generated	generated by, generated from
enhanced	enhanced in, enhanced by
activate	activates, activated
prevent	prevented, prevents, prevented by
play	plays
involve	involved in, involves
reveal	revealed, revealed between
detect	detected, detected in, detected by

The major idea of generating a semantic net is to highlight the role of a concept in a text corpus by eliciting its relationship to other concepts. The nodes in a semantic net represent entities and links indicate relationships. While concept ontologies are specialized types of semantic net, which also highlight the *taxonomical* and *partonomical* relations among concepts, the proposed semantic net is designed only to represent the generic biological relations and associated entities mined from the text corpus. Hence, a subset of an information component, termed as relation triplet, is used for this purpose. The relation triplet can be defined formally as follows:

Definition 2 (*Relation Triplet*). A relation triplet (RT) is a projection of information component which is defined as a triplet of the form $\langle S, V, O \rangle$, where V is a relational verb and S, O are noun phrases associated through V.

The whole graph is centered around a concept selected from the list of feasible concepts recognized by the *key concept miner* module. For a relation triplet $\langle S, V, O \rangle$, the biological entities present in S and O are used to define classes and V is used to define relationships between them. Since S and O may contain multiple biological entities, only the first entity identified by ABNER are displayed as class label in the semantic net for simplicity purpose. To define a relationship map, the user selects a concept, say ξ, around which the graph is to be created. The selected concept ξ is used to extract all those relation triplets which contains ξ either as a part of S or O or both. Hence for a relation triplet $\langle S, V, O \rangle$ three cases may arise:

Case 1: ξ *appears as a part of* S – In this case a separate node labeled with first entity appearing in S is created which is linked with a directed edge originating from ξ and labeled with V.

Case 2: *ξ appears as a part of O* – In this case a separate node labeled with first entity appearing as a part of *O* is created which is linked with a directed edge terminating at *ξ* and labeled with *V*.

Case 3: *ξ appears as a part of both S and O* – This combines both *case 1* and *case 2*.

Algorithm `SemanticNetGeneration` shown in Table 11 is used to convert the semantic net generation process into a working module. A snapshot of the semantic net generated around "*Alzheimer*" is shown in Fig. 5. The left pan of Fig. 5 shows the list of all feasible key concepts identified by *key concept miner* around which a semantic net can be generated. The user selected concept is displayed in oval at the center position and all related noun phrases containing at least one biological entity are displayed around it in rectangles. The color scheme is used to highlight the biological class of the associated entities. For visibility purpose, we have used the color scheme different from the one used by

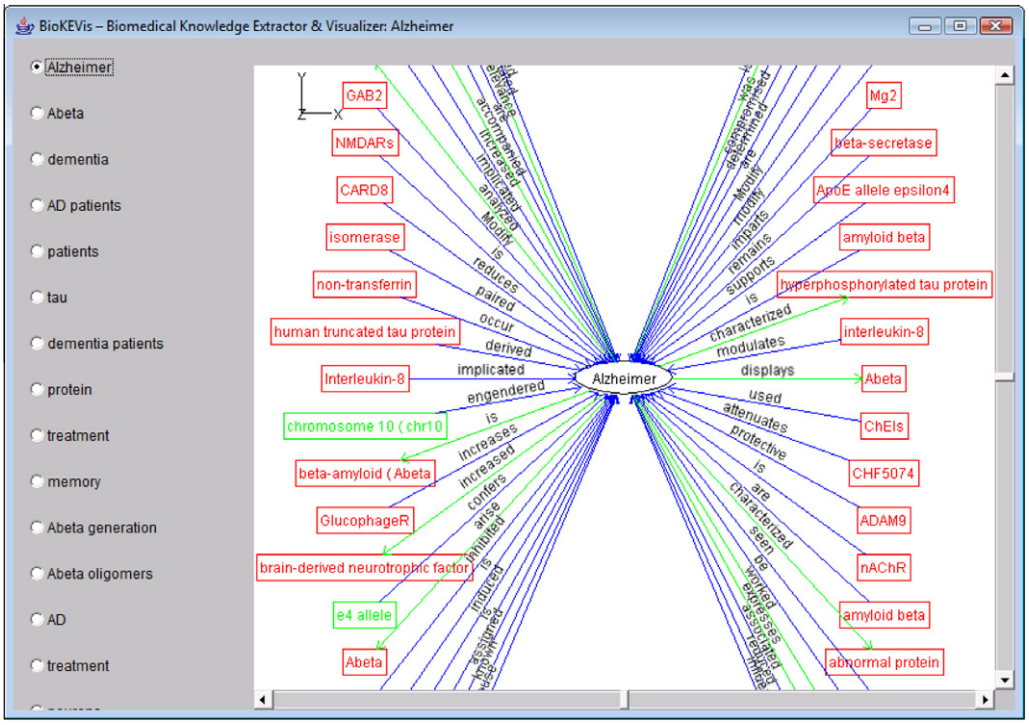

Fig. 5. Semantic net created by `BioKEVis` around *"Alzheimer"*.

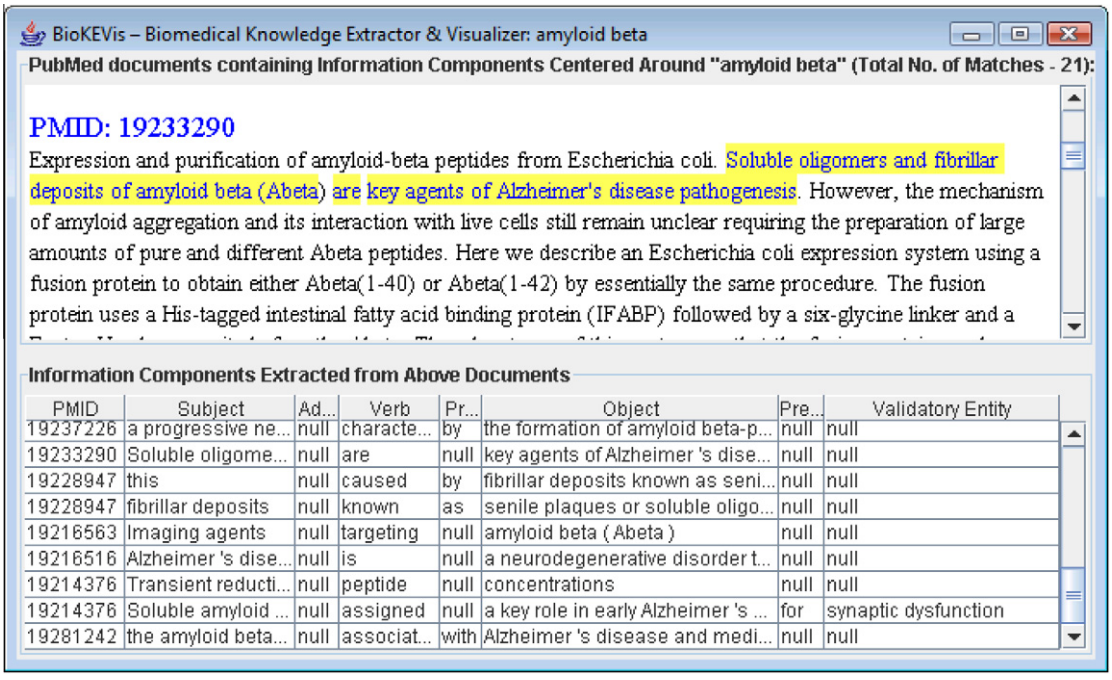

Fig. 6. PubMed documents and information components centered around "*amyloid beta*".

ABNER. In our visualization module "*red*" is used for *protein* class, "*green*" for *DNA* class, "*pink*" for *RNA* class, "*magenta*" for *cell-line* class and "*blue*" for *cell-type*. Since, the ABNER recognizes only a subset of GENIA ontology concepts – *protein*, *DNA*, *RNA*, *cell-line* and *cell-type*, at present the system highlights the class of only these entities appearing in the text.

The semantic net also facilitates the users to navigate through the pile of documents in an efficient way. While double-clicking on a node, all the information components (ICs) in which the entity, contained in the node, appears either as a part of *subject* or *object* are selected. Thereafter, the PubMed documents containing these ICs are displayed in which the relevant parts of the sentences are highlighted. The ICs that are present in the retrieved documents are also extracted and displayed separately in the bottom pan of the same window. Fig. 6 presents a snapshot of the window containing PubMed documents and information components centered around the entity "*amyloid beta*" when it was double-clicked in Fig. 5. Similarly, on double-clicking an edge, all information components (ICs) centered around the biological relation appearing as edge label are selected. Thereafter, the PubMed documents containing these ICs are displayed with properly highlighting the relevant snippet of text. The ICs that are present in retrieved documents are also extracted and displayed separately in bottom pan of the same window. Fig. 7 presents a snapshot of the window containing PubMed documents and information components centered around the relational verb "*modulates*" when it was double-clicked in Fig. 5.

3.6. Biomedical Query Processor

In this section, we present the design of the *Biomedical Query Processor* module, which processes user queries over the abstract database and displays relevant information components. The PubMed documents containing the information components (ICs) entered by user are displayed with properly highlighting the relevant snippet of text. The ICs that are present in the retrieved

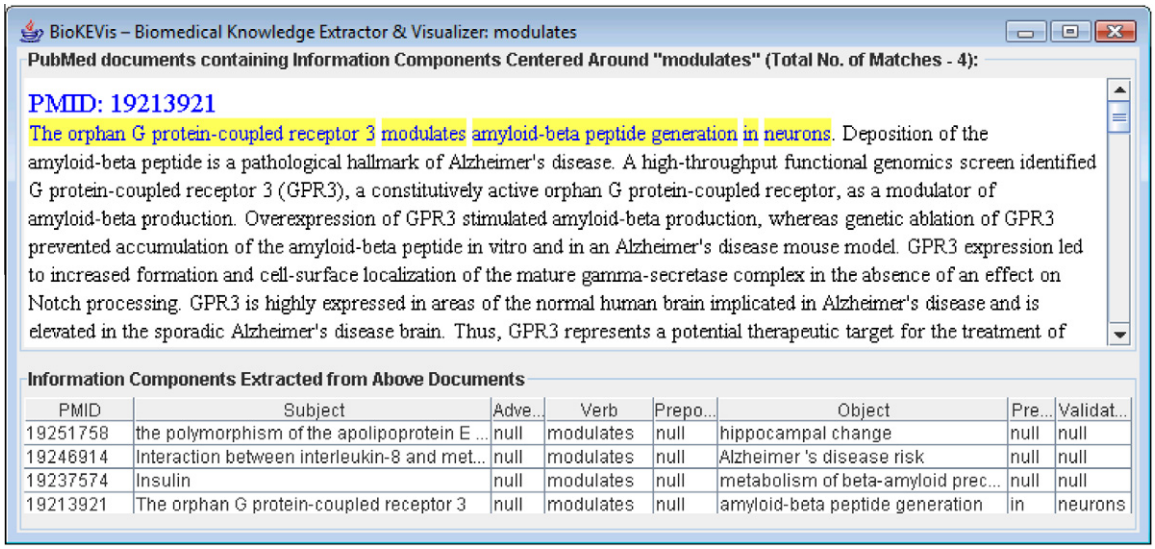

Fig. 7. PubMed documents and information components centered around "*modulates*".

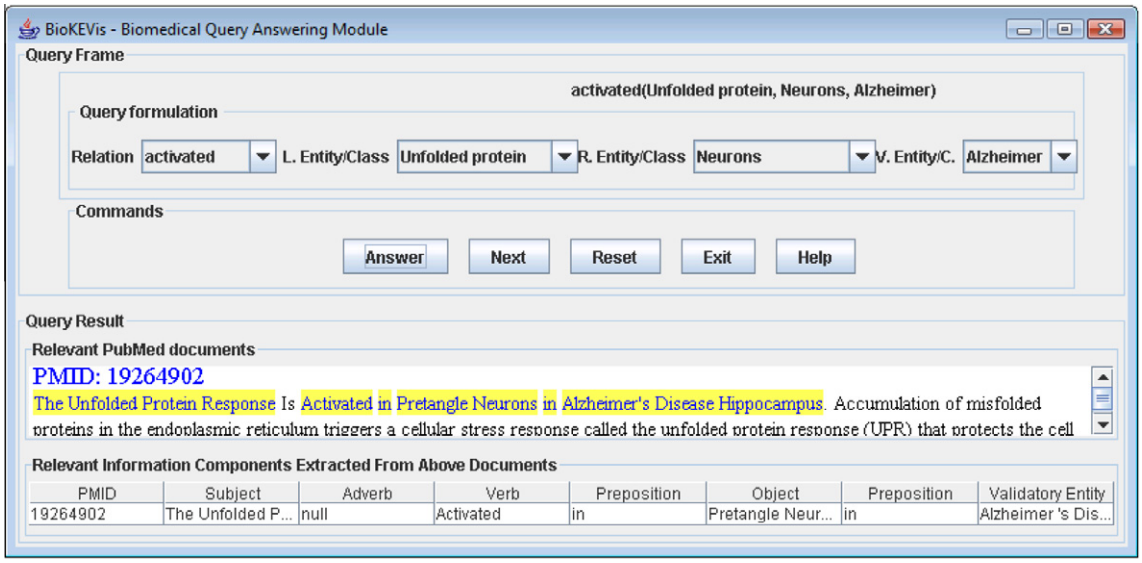

Fig. 8. Query interface.

documents are also extracted and displayed separately in the bottom pan of the query window. Links to PubMed abstracts are also created that can be used to navigate through the whole documents.

Query processing is a two-step process – acceptance and analysis of user query and finding relevant snippet of texts from the structured knowledge base. A query is represented by a template ⟨*leftEntity/class/**, *relation/**, *rightEntity/class/**, *validatoryEntity/class/**⟩ which allows the user to formulate feasible queries at multiple levels of specificity. A query can contain a mixture of concepts and entity names and/or a specific relation. The asterisk (*) symbol in any field represents a wild card entry and any match is considered as successful. A query is restricted to contain a maximum of three wild-card entries, since all four wild-card entries would be similar to retrieving all documents in the database. Fig. 8 shows a snapshot of the query interface and a partial list of sentences retrieved for the query ⟨*unfolded protein*, *activated*, *neurons*, *Alzheimer*⟩. Initially, the fields in the query interface contain all possible values from the corresponding constituents of the information components. When the user selects a specific value in a field, only the relevant elements for the remaining fields are displayed by the system. Thus guided query formulation allows users to specify only meaningful queries with respect to the underlying corpus.

Since the result set for a given query may contain a large number of documents, a relevance computation mechanism based on the associations of information components is introduced. For a given query the retrieved documents are displayed in non-increasing order of their degree of relevance to the query. The relevance value is calculated by using statistical based vector-space model. In this model a retrieved document d_j in response to a user query $\langle e_1, r, e_2, e_3\rangle$ is defined by a 4-dimensional vector $\vec{d}_j = (w_{e_1 j}, w_{r j}, w_{e_2 j}, w_{t j})$, where $w_{e_1 j}, w_{r j}, w_{e_2 j}$, and $w_{t j}$ represents the weights of e_1, r, e_2, and triplet $t = \langle e_1, r, e_2\rangle$, respectively in jth document. The weights are calculated using Eqs. (4)–(7). In Eq. (4), $tf_{e_1 j}$ represents the term-frequency of e_1 in jth document and $|\{\langle e_1, r, e_k\rangle : e_k \neq e_2\}|$ represents the number of entities, except e_2, associated with e_1 through r across the corpus. In Eq. (5), $tf_{r j}$ represents the term-frequency of r in jth document and $|\{\langle e_i, r, e_j\rangle : e_i \neq e_1 \wedge e_j \neq e_2\}|$ represents the number of entity-pairs, except $\langle e_1, e_2\rangle$, associated through r across the corpus. In Eq. (6), $tf_{e_2 j}$ represents the term-frequency of e_2 in jth document and $|\{\langle e_i, r, e_2\rangle : e_i \neq e_1\}|$ represents the number of entities, except e_1, associated with e_2 through r across the corpus. In Eq. (7), $tf_{t j}$represents the term-frequency of triplet t in jth document and $|\{\langle e_1, r_k, e_2\rangle\}|$ represents the number of relations associating the entity-pair $\langle e_1, e_2\rangle$ across the corpus. Finally, the degree of relevance of the document d_j with the user query, $rel(d_j)$, is calculated using Eq. (8). The relevance values calculated so are used to rank the retrieved documents in non-decreasing order of their relevance to user query

$$w_{e_1 j} = tf_{e_1 j} \times \log_2(|\{\langle e_1, r, e_k\rangle : e_k \neq e_2\}|) \tag{4}$$

$$w_{r j} = tf_{r j} \times \log_2(|\{\langle e_i, r, e_j\rangle : e_i \neq e_1 \wedge e_j \neq e_2\}|) \tag{5}$$

$$w_{e_2 j} = tf_{e_2 j} \times \log_2(|\{\langle e_i, r, e_2\rangle : e_i \neq e_1\}|) \tag{6}$$

$$w_{t j} = tf_{t j} \times \log_2(|\{\langle e_1, r_k, e_2\rangle\}|) \tag{7}$$

$$rel(d_j) = \sqrt{w_{e_1 j}^2 + w_{r j}^2 + w_{e_2 j}^2 + w_{t j}^2} \tag{8}$$

4. Experimental evaluation

The performance of the whole system is analyzed by taking into account the performance of the *key concept extraction* and *biological relation extraction* processes. For evaluation of the experimental results, we use standard Information Retrieval (IR) performance measures defined in Eqs. (9)–(11). From the extraction results, we calculate the true positive TP (number of correct concepts the system identifies as correct), the false positive FP (number of incorrect concepts the system falsely identifies as correct), and the false negatives FN (number of correct concepts the system fails to identify as correct). By using these values we calculate the following performance measures:

Precision (π): the ratio of true positives among all retrieved instances

$$\pi = \frac{TP}{TP + FP} \tag{9}$$

Recall (ρ): the ratio of true positives among all positive instances

$$\rho = \frac{TP}{TP + FN} \tag{10}$$

F_1-*measure* (F_1): the harmonic mean of *recall* and *precision*

$$F_1 = \frac{2\rho\pi}{\rho + \pi} \tag{11}$$

4.1. Evaluation of key concept extraction process

In this section we present a discussion on the performance of the key concept extraction module. For evaluation purpose we have used GENIA corpus [28] in which entity names are tagged with GENIA ontology concepts. Due to memory space limitation for using LSA function of MatLab, we have randomly taken only 50 documents from GENIA corpus for the evaluation purpose. A preprocessing module is implemented in Java that extracts all tagged entities and stores them in a list, say L. Then, it filters out all meta language tags from the documents. The filtered documents are parsed using Stanford parser to generate phrase structure which is later analyzed by *key concept miner* to identify feasible key concepts. Identified feasible concepts are ordered in non-increasing order of their weights shown in matrix M of Fig. 3(h). Thereafter, the concepts appearing at top 10%, top 20% and so on positions are considered for performance analysis. For each consideration, we have calculated the value of true positives (TP) and false positives (FP). Since false negative (FN) represents the entities in L that are not identified by the system as feasible concepts, for a partial list of extracted concepts by the system it

Table 11
Algorithm for semantic net generation.

Algorithm: SemanticNetGeneration(L_{RT}, ξ)

Input: Relation triplets (L_{RT}) and a key concept (ξ) around which the graph is to be created
Output: Semantic Net – A directed graph G = (V, E)

```
1.  V ← ξ
2.  E ← φ
3.  For all ⟨S,V,O⟩ ∈ L_IT do
4.    If ξ ∈ substring(S) then
5.      E_1 ← getFirstEntity(O)
6.      If E_1 ∉ V then
7.        V ← V ∪ E_1
8.        E ← E ∪ ⟨ξ, E_1⟩
9.      End if
10.   End if
11.   If ξ = substring(O) then
12.     E_1 ← getFirstEntity(S)
13.     If E_1 ∉ V then
14.       V ← V ∪ E_1
15.       E ← E ∪ ⟨E_1, ξ⟩
16.     End if
17.   End if
18. End for
19. Return G
```

Table 12
Misclassification matrix of the key concept extraction process.

Performance measure	Percentage of top position concepts considered as key concepts										
	10	20	30	40	50	60	70	80	90	100	F_1-measure
TP	37	69	104	132	174	203	259	302	351	393	0.68
FP	10	20	34	42	59	81	94	108	123	403	
FN	–	–	–	–	–	–	–	–	–	220	
Precision	0.79	0.78	0.75	0.76	0.75	0.71	0.73	0.74	0.74	0.73	
Recall	–	–	–	–	–	–	–	–	–	0.63	

Table 13
Performance evaluation of the biological relation extraction process.

Biological relations around which ICs are centered	Total # of times IC is identified by the system	Total # of times IC is correctly identified by the system	Total # of times IC occurs correctly in the test corpus	Precision (%)	Recall (%)	F_1-measure (%)
Activate	36	35	49	97.22	71.43	82.35
Associate	19	18	22	94.74	81.82	87.81
Express	26	24	35	92.31	68.57	78.69
Increase	19	17	26	89.47	65.38	75.55
Induce	71	67	91	94.37	73.63	82.72
Inhibit	36	34	48	94.44	70.83	80.95
Modulate	6	5	6	83.33	83.33	83.33
Reduce	22	21	30	95.45	70.00	80.77
Regulate	31	28	37	90.32	75.68	82.35
Stimulate	22	21	30	95.45	70.00	80.77
Average				92.71	73.07	81.53

would not be possible to decide the value of FN. So, the value FN is shown only for 100% consideration in Table 12.

Based on the values of TP, FP, and FN the *precision*, *recall* and F_1-*measure* values are calculated. Table 12 summarizes the performance measure values for our system in the form of a misclassification matrix. The recall value is lower than precision indicating that certain correct key concepts could not be recognized by the system correctly which leaves scope for enhancing our grammar to accommodate more dependency relations. Moreover, while calculating the values of TP, FP, and FN we have applied exact string matching which is also one of the reasons to lower these values.

4.2. Evaluation of relation extraction process

A relation triplet is said to be *correctly identified* if its occurrence within a sentence along with its left and right entities is grammatically correct and the system has been able to locate it in the right context. To judge the performance of the system, it is not enough to judge the extracted relations only, but it is also required to analyze all the correct relations that were missed by the system. The system was evaluated for its *recall* and *precision* values for 10 relations *activate*, *associate*, *express*, *increase*, *induce*, *inhibit*, *modulate*, *reduce*, *regulate*, and *stimulate*. Like evaluation of key concept extraction module, an evaluation software was written in Java for this module too which exhaustively checks the corpus for possible occurrences of the required relation. For each relation to be judged, the evaluation software takes the root relation as input and performs partial string matching to extract all possible occurrences of the relation. This ensures that various nuances of English language grammar can also be taken care of. For example, if the root relation used in any query is "*activate*", all sentences containing *activates*, *inactivate*, *activated by*, *activated in*, *etc.* are extracted. Each sentence containing an instance of the pattern is presented to the human evaluator after its appropriate tagging through ABNER. The sentence without ABNER tags is also presented to the evaluator. This makes it easier for the evaluator to judge the grammatical correctness of the relation in association to the concepts or entities around it. Each occurrence of the relation is judged for correctness by the evaluator, and the correct instances are marked. The marked instances are stored by the evaluation software and later used for computing the precision and recall values.

The precision value of the system reflects its capability to identify a relational verb along with the correct pair of concepts/entities within which it is occurring. Recall value reflects the capability of the system to locate all instances of a relation within the corpus. Table 13 summarizes the performance measure values of our relation extraction system in the form of a misclassification matrix for information components centered around 10 different biological relations. On 100 documents randomly selected from GENIA corpus, the average precision, recall, and F_1-measure values are 92.71%, 73.07%, and 81.53% respectively.

As is observed, the precision of the system is quite high. This indicates that most of the extracted instances are correctly identified. However, the recall value of the system is somewhat low. This indicates that several relevant elements are not extracted from the text. The reason for low recall values was identified as follows. We observed that most miss occur when the parser assigns an incorrect syntactic class to a relational verb. For example, in the following sentence, the relational verb "activates" and other related constituents could not be identified by the system because "activates" is marked as noun by the parser. Similarly, other misses occur when an information components spans over multiple sentences using anaphora.

''Increased [Ca2+]i activates Ca2+/calmodulin-dependent kinases including the multifunctional Ca2+/calmodulin-dependent protein kinase II (CaM-K II), as well as calcineurin, a type 2B protein phosphatase [MEDLINE #: 95173590, Sentence No. 2].''

5. Uniqueness of the proposed framework

In this section, we highlight some of the key features of the proposed system BioKEVis over the existing systems in literature. The results presented in the previous sections are comparable to

the results of other methods in literature, but we can note that the tasks are not the same. Unlike most of the related works [9–12] on biological relation extraction, which have described methods for mining a *fixed set* of biological relations occurring with a set of predefined tags, the proposed system identifies all verbs in a document, and then identifies the feasible biological relational verbs using contextual analysis. While mining biological relations the associated prepositions are also considered which very often changes the nature of the verb. For example, the relation "*activates in*" denotes a significant class of biological reactions. Thus, we also consider the biological relations, which are combinations of *root verbs, morphological variants, and prepositions* that follow these. Typical examples of biological relations identified in this category include "*activated in*", "*binds to*", "*stimulated with*", *etc.* Besides mining relational verbs and associated entities, the novelty of the system lies in extracting *validatory entities* whose presence or absence validates a particular biological interaction. BioKEVis also extracts the adverbs associated with relational verbs, which plays a very important role especially to identify the negation in sentences that are very crucial while answering user queries.

We have also presented a scheme for semantic integration of information extracted from text documents using semantic net which highlights the role of a single entity in various contexts. The network provides distinct user perspectives and allows navigation over documents with similar information components and is also used to provide a comprehensive view of the collection. The integration of the system with biological entity (*DNA*, *RNA*, *protein*, *cell-line*, and *cell type*) recognizer helps in answering queries formulated at different levels of specificity. Given a query, BioKEVis aims at retrieving all relevant sentences that contain a set of biological concepts stated in a query, in the same context as specified in the query, from the curated database. The document ranking mechanism to present retrieved documents in order of their relevance to the user query is also unique over existing biomedical query answering systems like MedTAKMI [1] and Textpresso [2].

6. Conclusion and future work

In this paper, we have proposed the design of a novel biomedical knowledge extraction and visualization system, BioKEVis, for conceptualization of document corpora and biomedical query answering. The system uses linguistic and semantic analysis of text to identify key information components from biomedical text documents and stores them in a structured knowledge base over which biomedical queries are processed. The information components are centered on domain entities and their relationships, which are extracted using natural language processing techniques and co-occurrence-based analysis. The system is also integrated with a biomedical entity recognizer, ABNER, to identify a subset of GENIA ontology concepts (*DNA*, *RNA*, *Protein*, *Cell-line*, and *Cell-type*) in the texts and tag them accordingly. This helps in answering queries based on biological concepts rather than on particular entities only.

We have also proposed a method for collating information extracted from multiple sources and present them in an integrated fashion with the help of semantic net. The semantic net highlights the role of a single entity in various contexts which are useful both for a researcher as well as a layman. One of the unique features of our system lies in its capability to mine and extract information about generic biological relations and the associated prepositions from biomedical text documents. The system also extracts validatory entities associated with relation triplets which presence or absence validates biological interactions. This is also a unique aspect of BioKEVis over other existing approaches. The system is integrated with a query-processing module that allows users to formulate queries in a guided way at different levels of specificity.

Since the system advocates using biological relations in queries, the information overload on the users can be substantially reduced. Right now the system uses only a subset of GENIA ontology concepts. In future, we are planning to train the biological entity recognizer, ABNER, on GENIA corpus to make it capable to recognize all GENIA ontology concepts in a plain text. The relation extraction rules are also being refined to improve the *precision* and *recall* values of the system. Moreover, the design of the query processing module is being enhanced to handle more complex biomedical queries in an efficient way.

References

[1] Uramoto N, Matsuzawa H, Nagano T, Murakami A, Takeuchi H, Takeda K. A text-mining system for knowledge discovery from biomedical documents. IBM Syst J 2004;43(3):516–33.

[2] Muller HM, Kenny EE, Strenber PW. Textpresso: an ontology-based information retrieval and extraction system for biological literature. PloS Biol 2004;2(11):e309. Available from: <http://www.plosbiology.org> .

[3] Shatkay H et al. Information retrieval meets gene analysis. IEEE Intell Syst 2002;17:45–53.

[4] Allen J. Natural language understanding. 2nd ed. Indian branch: Pearson Education (Singapore) Pvt. Ltd.; 2004.

[5] Schuler GD et al. Entrez: molecular biology database and retrieval system. Methods Enzymol 1996;266:141–62.

[6] Cowie J, Lehnert W. Information extraction. Comm Assoc Comput Mach 1996;39:80–91.

[7] Riloff E, Lehnert W. Information extraction as a basis for high-precision text classification. ACM Trans Inform Syst 1994;12:296–333.

[8] Ding J, et al. Mining medline: abstracts, sentences or phrases. In: Proceedings of the pacific symposium on biocomputing, 2002. p. 326–37.

[9] Sekimizu T, Park HS, Tsujii J. Identifying the interaction between genes and genes products based on frequently seen verbs in Medline abstract. Genome Inform 1998;9:62–71.

[10] Thomas J, Milward D, Ouzounis C, Pulman S, Carroll M. Automatic extraction of protein interactions from scientific abstracts. In: Pacific symposium on biocomputing, 2000. p. 538–49.

[11] Ono T, Hishigaki H, Tanigami A, Takagi T. Automated extraction of information on protein–protein interactions from the biological literature. Bioinformatics 2001;17(2):155–61.

[12] Rinaldi F, Scheider G, Andronis C, Persidis A, Konstani O. Mining relations in the GENIA corpus. In: Proceedings of the 2nd European workshop on data mining and text mining for bioinformatics, Pisa, Italy, 2004.

[13] Settles B. ABNER: an open source tool for automatically tagging genes, proteins, and other entity names in text. Bioinformatics 2005;21(14): 3191–2.

[14] Schutz A, Buitelaar P. RelExt: a tool for relation extraction from text in ontology extension. In: International semantic web conference, 2005. p. 593–606.

[15] Yakushiji A, Teteisi Y, Miyao Y, Tsujii J. Event extraction from biomedical papers using a full parser. In: Pac Symp Biocomput, 2001. p. 408–409.

[16] Cox E. A hybrid technology approach to free-form text data mining. Scianta Intelligence, 2005. *URL*: <http://scianta.com/pubs/AR-PA-007.htm>.

[17] Castro AG et al. The use of concept maps during knowledge elicitation in ontology development processes – the nutrigenomics use case. BMC Bioinform 2006;7(267).

[18] Ciaramita M, Gangemi A, Ratsch E, Saric J, Rojas I. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In: Proceedings of the 19th international joint conference on artificial intelligence (IJCAI'05), 2005. p. 659–64.

[19] Feldman R, et al. Maximal association rules: a new tool for mining for keyword co-occurrences in document collections. In: Proceedings of the 3rd international conference on knowledge discovery, Newport Beach, CA, USA, 1997. p. 167–70.

[20] Aumann Y, et al. Circle graphs: new visualization tools for textmining. In: Proceedings of the 3rd European conference on principles and practice of knowledge discovery in databases, 1999. p. 277–82.

[21] Jenssen T-K, Laegreid A, Komorowski J, Hovig E. A literature network of human genes for high-throughput analysis of gene expression. Nat Genet 2001;28:21–8.

[22] Gaizauskas R, Demetriou G, Artymiuk PJ, Willett P. Protein structures and information extraction from biological texts: the PASTA system. Bioinformatics 2003;19(1):135–43.

[23] Friedman C, Kra P, Yu H, Krauthammer M, Rzhetsky A. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics 2001;17(1):s74–82.

[24] Zheng H-T, Borchert C, Kim H-G. Exploiting gene ontology to conceptualize biomedical document collections. LNCS, vol. 5367. Berlin/ Heidelberg: Springer; 2008. p. 375–89.

[25] Plaisant C, Fekete J-D, Grinsteinn G. Promoting insight-based evaluation of visualizations: from contest to benchmark repository. IEEE Trans Visual Comput Graph 2008;14(1):120–34.
[26] Landauer T, Foltz P, Laham D. Introduction to latent semantic analysis. Discourse Process 1998;25:259–84.
[27] Zheng H-T, Borchert C, Kim H-G. A concept-driven automatic ontology generation approach for conceptualization of document corpora. In: Proceedings of the 2008 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, vol. 01, 2008. p. 352–58.
[28] Kim J-D, Ohta T, Tateisi Y, Tsujii J. GENIA corpus – a semantically annotated corpus for bio-textmining. Bioinformatics 2003;19(1):i180–2.
[29] Salton G, McGill MJ. Introduction to modern information retrieval. New York, USA: McGraw-Hill, Inc.; 1986. ISBN:0070544840.
[30] Porter MF. An algorithm for suffix stripping. Program 1980;14(3):130–7.
[31] Berry MW, Hendrickson B, Raghavan P. Sparse matrix reordering schemes for browsing hypertext. In: Renegar J, Shub M, Smale S, editors. Lecture in applied mathematics, vol. 32, 1996. p. 99–123.
[32] Abulaish M, Dey L. Biological relation extraction and query answering from medline abstracts using ontology-based text mining. Data Knowl Eng 2007;61(2):228–62.
[33] Fundel K, Kuffner R, Zimmer R. RelEx – relation extraction using dependency parse trees. Bioinformatics 2007;23:365–71.
[34] Li J, Zhang Z, Li X, Chen H. Kernel-based learning for biomedical relation extraction. J Am Soc Inform Sci Technol 2008;59(5):756–69.