Article

JIS

Journal of Information Science
1–18

DOI: 10.1177/0165551513494645
jis.sagepub.com
SAGE

Concept map construction from text documents using affinity propagation

Iqbal Qasim
Hanyang University, South Korea

Jin-Woo Jeong
Hanyang University, South Korea

Jee-Uk Heu
Hanyang University, South Korea

Dong-Ho Lee
Hanyang University, South Korea

Abstract
Concept maps are playing an increasingly important role in various computing fields. In particular, they have been popularly used for organizing and representing knowledge. However, constructing concept maps manually is a complex and time-consuming task. Therefore, the creation of concept maps automatically or semi-automatically from text documents is a worthwhile research challenge. Recently, various approaches for automatic or semi-automatic construction of concept maps have been proposed. However, these approaches suffer from several limitations. First, only the noun phrases in text documents are included without resolution of the anaphora problems for pronouns. This omission causes important propositions available in the text documents to be missed, resulting in decreased recall. Second, although some approaches label the relationship to form propositions, they do not show the direction of the relationship between the subject and object in the form of Subject–Relationship–Object, leading to ambiguous propositions. In this paper, we present a cluster-based approach to semi-automatically construct concept maps from text documents. First, we extract the candidate terms from documents using typed dependency linguistic rules. Anaphoric resolution for pronouns is introduced to map the pronouns with candidate terms. Second, the similarities are calculated between the pairs of extracted candidate terms of a document and clusters are made through affinity propagation by providing the calculated similarities between the candidate terms. Finally, the extracted relationships are assigned between the candidate terms in each cluster. Our empirical results show that the semi-automatically constructed concept maps conform to the outputs generated manually by domain experts, since the degree of difference between them is proportionally small based on a Likert scale. Furthermore, domain experts verified that the constructed concept maps are in accordance with their knowledge of the information system domain.

Keywords
affinity propagation; concept map; concept map learning; knowledge acquisition; text clustering

1. Introduction

A concept map (CM) is a graphical node-arc representation that is widely used for organizing and representing knowledge and shows the relationships among related concepts. The nodes of a CM contain concepts, usually represented as nouns or noun phrases, and the links between the nodes capture the relationships among the concepts. The labelling of the links with a verb or a verb phrase allows the creation of propositions in a form of concept-label-concept chain [1].

Corresponding author:
Dong-Ho Lee, KDE Laboratory, Department of Computer Science and Engineering, ERICA Campus, Hanyang University, South Korea.
Email: dhlee72@hanyang.ac.kr

An individual concept does not provide meaning; however, when two concepts are associated using a link with a verb or verb phrase, they form a meaningful propositions [2].

Many educators and researchers have exploited CMs in a number of ways, including evaluation or assessment tools [3, 4], cooperative meaningful learning tools in education [5, 6], and advance organizer and visualization tools [7]. Also, CMs have been used as a means for communicating information for organizing ideas and promoting problem solving strategies [8, 9]. A CM can be used as an intermediate step in ontology learning for a particular domain to support the acquisition of domain knowledge [10].

The automatic or semi-automatic extraction of CMs from text documents is called concept map mining (CMM) [11]. It consists of extracting a list of concepts from a text and determining the linking relationship that should connect to form meaningful propositions. The extracted CM should represent a generic summary of a text document. Current concept map mining techniques can be classified into the following three main categories: (1) statistical; (2) linguistic; and (3) hybrid. The statistical approaches rely on techniques that are based on quantitative indicators and metrics value (e.g. frequency of occurrence and clustering techniques) [12]. Although statistically based approaches are simple to handle documents independently [13], they suffer from unpredictable results and semantic loss during document handling [14]. In linguistic-based approaches, results rely on techniques based on computational linguistics [12]. They use syntactic and semantic patterns, pragmatics and discourse resources to handle the text documents [13]. Although purely linguistic-based approaches are more accurate than statistical approaches, they are based on external knowledge databases, such as dictionaries, thesaurus or lexical data sources [14]. Hybrid approaches offer a combination of statistical and linguistic approaches, based on syntactic parsing, linguistic filters and statistical measures. Hybrid approaches increase the recall and precision, while reducing the noise in extracted propositions for a particular domain. However, these approaches suffer from a limitation of high computational cost.

In this paper, we present a new method to construct a concept map from unstructured text for knowledge extraction and representation. First, we extract all the candidate terms from a document using typed dependency linguistic rules. Then the anaphora resolution problem for pronouns is handled using the Resolution of Anaphore Procedure (RAP) algorithm [15]. Second, the similarities are calculated between pairs of extracted candidate terms of a document based on the assumption that, if the frequency of co-occurrence between the two terms is high, then the relation between them is also high. Then we exploit the affinity propagation algorithm [16], which takes as input the calculated similarities between pairs of extracted candidate terms to define the clusters. Finally, we extract the taxonomic and non-taxonomic relationships and assign them between the pairs of candidate terms in each cluster.

The rest of this paper is organized as follows: Section 2 presents related work and a brief overview of affinity propagation clustering algorithm. Section 3 describes the proposed hybrid approach for concept map mining from a text document. Section 4 explains experimental results and discussion on selected documents from the information system domain. Finally, conclusions are drawn in Section 5.

2. Literature review

In this section, we briefly describe the related work on concept map mining from text documents. Section 2.2 introduces the affinity propagation algorithm and explains the clustering process using affinity propagation to provide a better understanding of our approach.

2.1. Related work

In the literature, the area of study addressed by this paper is called domain concept map construction or domain concept map mining. In this approach, a process of extracting concepts and relationships (taxonomic and non-taxonomic) is carried out that connects the concepts to form meaningful propositions from an extensive amount of unstructured text for a particular domain. In recent years, there have been many research studies focused on the development of techniques to extract concept maps from existing text documents for a particular domain. A number of systems have recently been proposed for concept map learning from text documents. We describe some of them subsequently.

More recently, Chen et al. [17] extracted concept maps by mining academic articles. In this work, four main steps were performed: first, information retrieval techniques were applied and concept items were extracted. Keywords available in academic articles were used as concept candidates. Afterwards, research keyword indexing was performed using principle component analysis to select representative research keywords. Second, the relation strength between two keywords was calculated on the basis of their degree of correlation, taking into account the number of times that two keywords co-occurred in the same sentence. Finally, query parameters were used to extract concept maps through a user friendly interface. The evaluation was performed manually by inviting two experts in the domain of the corpus who assessed the quality of the maps. They evaluated the quality of the 30 most important concepts and the relationships between them. They agreed that the relationship between the concepts were 80% compliant with their professional knowledge where as the ranking of concepts was 70% compliant. However, Villalon and Calvo [11] pointed out an issue where the lack of an evaluation framework to measure the quality of the generated maps among different techniques and the relationship between the concepts was not labelled in the work.

Tseng et al. [18] proposed a heuristic algorithm to automatically construct a concept map from learners' historical testing records. They divided their methodology into two phases. In phase 1, all testing records were preprocessed using fuzzy association rule mining. In phase 2, the mined association rules were transformed into a prerequisite relationship between the extracted concepts to construct a concept map. The resulting concept map could be used to assist misconception diagnosis, adaptive learning and customized course design. The authors provided a series of interesting examples to clarify this method. However, no experiment was conducted to evaluate the quality of the constructed concept map. Zouaq and Nkambou [10] presented a system called TEXCOMON that constructs domain concept maps from textual resources and then transforms these concept maps into domain ontologies. Clariana and Koul [19] proposed a term association method that relies on a predefined list of domain-specific concepts provided by an expert. It considers two concepts to be related if they occur in the same sentence; however, it does not suggest the possible relationships between the pairs of concepts. Therefore, propositions are not formed. Valerio and Leake [20] also used verbal and noun phrases to extract concepts and relationships; however, no hierarchy was extracted. Some other researchers have also automatically created concept maps with connected concepts; however, the relationships were not labelled [21–25].

These approaches made an effort to more accurately mine text documents and construct the concept maps. However, they still have some limitations. First, these approaches consider only noun phrases in the text documents without resolving anaphora resolution problems, which leads to missing important propositions available in the text documents. Anaphora resolution is the problem of resolving what a pronoun or a noun phrase refers to. Second, some of the approaches label the relationship to form propositions, but they do not show the direction of the relationship between subject and object in the form of Subject–Relationship–Object, leading to ambiguous propositions. Therefore, the coverage of mined knowledge from particular text documents may not be satisfactory. In this paper, we propose an efficient concept map mining approach in which more propositions are discovered by including the pronominal anaphora resolution problem. In addition, accurate propositions are discovered by determining the subject and object of a sentence and assigning taxonomic and non-taxonomic relationships between them.

2.2. Affinity propagation

Motivated by the idea that concept map construction from text documents is a clustering problem in nature, we adopt an effective and scalable clustering technique, affinity propagation (AP) [16], for clustering the extracted candidate terms from text documents. To ensure that it is self-contained, this section describes affinity propagation in detail.

Affinity propagation is a powerful clustering technique based on the message passing proposed in Frey and Dueck [16]. AP identifies the set of exemplars that represent all the data points well in a data set. Initially, a square matrix of similarity values among the data points is provided as an input to the algorithm. First, all the data points are considered as candidate exemplars. Exemplars that emerge as real-valued messages are then passed between data points and each data point is assigned to an exemplar. Here, we describe the mathematical model of the AP approach to provide a better understanding of our approach. Initially, AP takes in input from a collection of real-valued similarities between data points. Given a data set with N data points (i.e. candidate terms of a domain), x_i and x_j are two objects (candidate terms) in it. The similarity $s(i,j)$ indicates how similar data point x_i is to another data point x_j. For instance, the cosine coefficient can be used to measure the similarities among all the data points. However, we use the structural similarity between each pair of data points with respect to the text documents because the cosine coefficient does not cover the context in which both the candidate terms are used in a document.

In the following section, we will explain the method of computing the structural similarity measures in more detail. $s(i,i)$ evaluates the self-similarity also called the 'preference' [16], and it is often set as a common value. This value can be varied to produce a different number of clusters. If the preference value is high then the number of exemplars and clusters is also high. The shared (preference) value can be the median of the input similarities (resulting in a moderate number of clusters) or their minimum (resulting in a small number of clusters). Also, there are two types of messages exchanged between the data points called responsibility and availability. In responsibility $r(i,j)$, a message is sent from data point i to candidate exemplar point j, and reflects the accumulated evidence in terms of the real value for how well-suited point j is to serve as the exemplar for point i. In availability $a(i,j)$, a message is sent from the candidate exemplar point j to point i, and reflects the accumulated evidence for how appropriate it would be for point i to choose point j as its exemplar. Figure 1(a) depicts how responsibility $r(i,j)$ messages are sent from data points (candidate terms) to

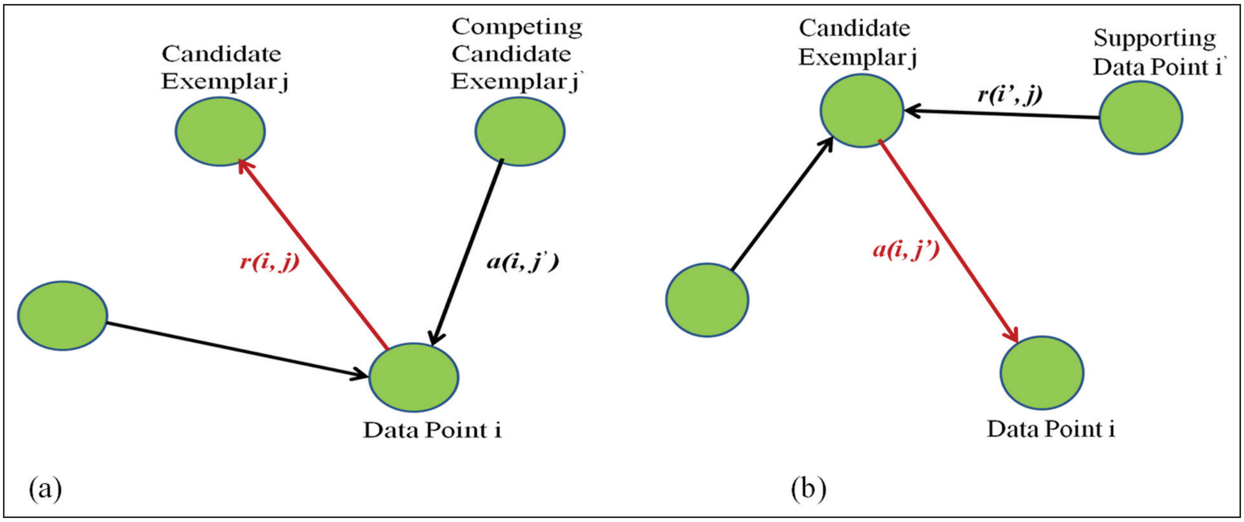


Figure 1. Sending responsibilities and availabilities [16]. (a) Sending responsibilities. (b) Sending availabilities.

exemplars (candidate terms) and indicates how each data point favours the exemplar over the other candidate exemplars. Figure 1(b) depicts how availability $a(i,j)$ messages are sent from the candidate exemplars (candidate terms) to the data points (candidate terms) and shows how appropriate each candidate exemplar is as a cluster centre for the data points (candidate term). At the first iteration, the availability messages are initialized to zero: $a(i,j)=0$. The responsibilities are then computed using the responsibility update equation formulated as:

$$r(i,j)=s(i,j)-\max_{j'\neq j}\{a(i,j')+s(i,j')\} \tag{1}$$

In the first iteration, because the availabilities between all the data points are zero, $r(i,j)$ is set to the input similarity between point i and point j, minus the maximum of the similarities between point i and the other candidate exemplars. When the above responsibility update lets all the candidate exemplars compete for ownership of a data point, the availabilities are computed using the availability update equation written as:

$$a(i,j)=\min\left\{0,r(j,j)+\sum_{i'\neq i,j}\max\left\{0,r\left(r^{i},j\right)\right\}\right\},i\neq j \tag{2}$$

Equation (2) shows that availability $a(i,j)$ is set to the minimum value between 0 and self-responsibility $r(j,j)$ plus the sum of the positive responsibilities received by the candidate exemplar j from the other data points. The self-availability $a(j,j)$ is updated differently using Equation (3):

$$a(j,\,j)=\sum_{i'\neq i}\max\{0,\;r(i',j)\},\;i=j \tag{3}$$

Equation (3) shows the accumulative evidence that point j is an exemplar based on the positive responsibilities sent to candidate exemplar j from the other data points. In addition, a damping factor $\lambda \in [0,1]$ is added during the message passing between the data points to avoid numerical oscillations that may arise in some circumstances. At any point during AP, the responsibilities and availabilities can be combined to identify the exemplars:

$$c_j \leftarrow \operatorname*{argmax}_{1\leq j\leq N}[r(i,j)+a(i,j)] \tag{4}$$

In brief, the AP process in each iteration consists of (1) updating all the responsibilities given the availabilities, (2) updating all availabilities given the responsibilities, and (3) combining the availabilities and responsibilities to monitor the exemplar decisions and terminate the AP algorithm when these decisions do not change for a few iterations. A more detailed explanation of the AP approach is available in Frey and Dueck [16].

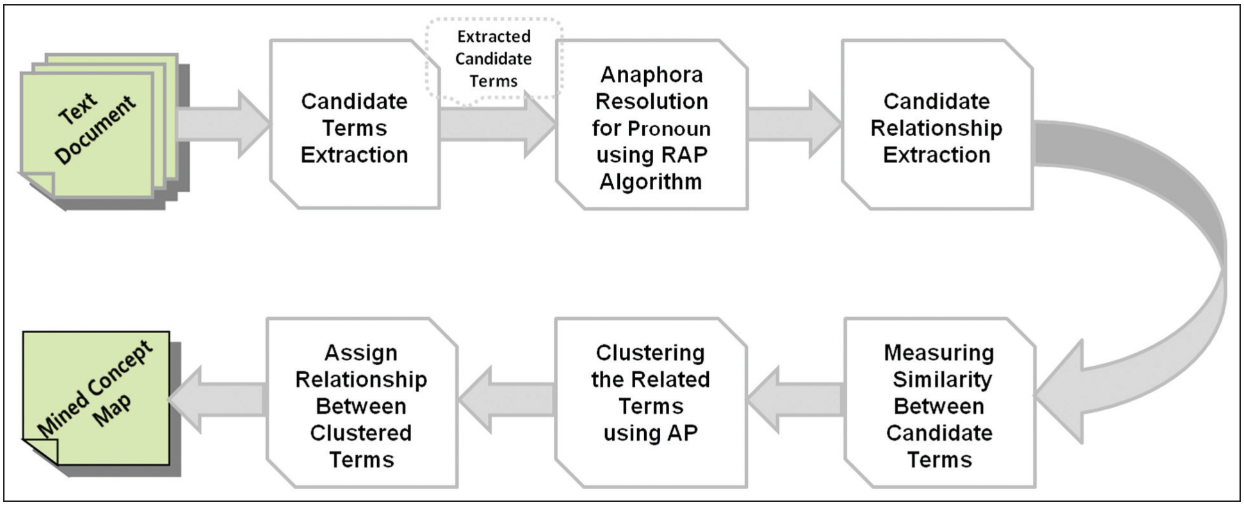


Figure 2. Proposed system methodology.

3. System design and methodology

In this section, we present how the proposed system extracts the candidate terms, maps the relationships between them and constructs the concept map using AP. As shown in Figure 2, there are six main steps in the procedure that are described in the following subsections.

3.1. Candidate terms extraction

Terms extraction refers to the discovery of all possible terms that are considered as potential candidate concepts for a concept map. According to Covington [26], the structure of natural language sentences is based on two grammar types: *constituency* and *dependency*. *Constituency grammar* describes phrase–structure syntax whereas, in *dependency grammar*, each word pair is related by a grammatical link called dependency. We exploit dependency grammar because grammatical link dependencies are intuitively close to semantic relationships, as described in Zouaq and Nkambou [10]. Several analysers can perform dependency analysis; however, the results presented in Zouaq and Nkambou [10] and Stevenson and Greenwood [27] suggest that the Stanford Parser [28] can generate a more accurate analysis. As an example, Table 1 represents the complete chart of typed dependencies for a given sample sentence using the Stanford Parser. These typed dependencies in Table 1 map onto a directed graph representation, in which words in the sample sentence are nodes in the grap whereas grammatical relations are edge labels. Table 1 shows simple and uniform representation which is quite accessible to non-linguistic thinking about tasks involving information from text, and it is quite effective for extraction of candidate terms and relationships between them. Column 1 shows the sample sentence and column 2 shows the extracted corresponding typed dependencies relationships of a given sample sentence in Table 1. The starting values show the typed dependency relationships leading the *governor* terms with its index number and then *dependent* terms with its index number in a given sentence are given. For example, in ‘nn(Systems-4, Intelligent-2)’, ‘nn’ is the typed dependency relationship, ‘Systems-4’ is the governor term with its index number, and ‘Intelligent-2’ is the dependent term with its index number in a given sentence.

After parsing, a corpus is established by applying the set of proposed patterns in Algorithm 1 to extract multi and single-word candidate terms by exploiting the type dependency graph. The term ‘pattern’ is a formalization of both the general principles and the idiomatic solutions used to extract the semantic relationship between the candidate terms in natural language. For example, applying linguistic patterns allows for the extraction of the candidate terms available in the sample sentence used in Table 1, which are ‘Intelligent Tutoring System’, ‘Educational Support System’ and ‘Class Oriented Education’. Table 2 provides some of the typed dependencies defined for the English. Typed dependencies presented in Table 2 were designed in the Stanford Parser to provide a simple description of the grammatical relationships in a sentence. These typed dependencies can easily be understood and effectively used by people without linguistic expertise to extract candidate terms and candidate relationships between these candidate terms. The dependencies presented in Table 2 are all in binary relations: a grammatical relation holds between a governor term and a dependent term. Column

Table 1. Extracted typed dependency using the Stanford Parser for the sentence: 'The intelligent tutoring systems are educational support systems, which improve general class oriented education.'

Example sentence	Extracted typed dependencies
The intelligent tutoring systems are educational support systems, which improve general class oriented education.	det(Systems-4, The-1) nn(Systems-4, Intelligent-2) nn(Systems-4, Tutoring-3) nsubj(systems-8, Systems-4) cop(systems-8, are-5) amod(systems-8, educational-6) nn(systems-8, support-7) root(ROOT-0, systems-8) nsubj(improve-11, systems-8) rcmod(systems-8, improve-11) amod(education-16, general-13) nn(education-16, class-14) nn(education-16, oriented-15) dobj(improve-11, education-16)

Table 2. Used typed dependency patterns with their examples.

Dependency symbol	Type	Example	Extracted Term
amod	Adjectival modifier	amod(commerce-20, electronic-19)	Electronic commerce
nn	Noun compound modifier	nn(web-13, semantic-12)	Semantic Web
advmod	Adverbal modifier	advmod(learning-11, ontology-10)	Ontology learning
prep_of	Of prepositional modifier	prep of(domains-12, interest-14)	Domains of interest
conj_and	Conjunct and	(database-1, philosophy-3)	Database and philosophy
nsubj	Nominal subject	(emerged-5, ontology-3)	Ontology

1 in Table 2 shows the abbreviated name of the typed dependency; column 2 shows the complete name of typed dependency; and columns 3 and 4 show the extracted terms from these dependency relations shown in Table 1.

In English, 'amod', 'nn', 'advmod', 'prep of', 'conj and' and 'nsub' are some of the typed dependencies. An adjectival modifier (*amod*) is any adjectival phrase that modifies the meaning of natural phrase; a noun compound modifier (*nn*) is any noun that modifies the head noun; an adverbial modifier (*advmod*) of a word modifies the meaning of the word; a prepositional modifier (*prep of*) modifies the meaning of the noun, verb or adjective; a conjunct (*conj and*) is the relation between two elements connected by a coordinating conjunction; and a nominal subject (*nsub*) is a noun phrase that is the syntactic subject of a clause. Some of the patterns are defined in Algorithm 1.

To summarize Algorithm 1, we start by extracting all typed dependencies in the domain documents using the Stanford Parser. We then extract only five types of dependency relationships (shown in Algorithm 1) from the extracted typed dependencies. All information related to dependency relationships is stored in different variables. Finally, we define some rules to extract candidate terms for the construction of a concept map. Table 3 shows the sample of extracted candidate terms with different lengths.

3.2. Anaphoric resolution for pronouns

Anaphora resolution is the process of determining and assigning each anaphor (pronoun) to the corresponding antecedent (noun). Linguistic patterns only extract simple noun phrases; however, complex relationships (main verb) exist between these pronouns and many propositions are missed during the construction of a concept map, causing a significant drop in recall. Therefore, pronominal anaphoric resolution is applied based on the RAP algorithm [15] to extract more complex propositions for the construction of a concept map. The RAP algorithm is applied to the syntactic structures generated by McCords Slot Grammar parser [29]. The RAP relies on the salience measures derived from syntactic structure and a simple dynamic model of the attentional state to select the 12 antecedent noun phrases of a pronoun from a list of candidates. A more detailed explanation of the RAP algorithm is available in [15]. As an example, Table 4 represents a conceptual link between the pronouns and nouns for a given sample text excerpt using the RAP algorithm.

Algorithm 1. Candidate terms extraction

Input: Natural language sentences $S \in D$ where D is set of domain documents
Output: Set of candidate terms.

```
/* Define variables and Extraction of type dependencies one by one */
pre_gov_index ← null
String candidate_term, prev_gov_term
Vector candidate_term _vector
typedependency_vector ← execute.stanfordparser(S)
/* Repeat the below statements until last value of type dependency. */
for (i = 0; i < typedependency_vector.length(); i++)
/* Extract 5 type dependency relationships (amod, nn, conj and, prep of, nsubj)one by one.*/
relationship ← execute.extractrelationships(dependencies_vector[i])
/* Find dependent and governor term and store. */
dep_term ← execute.getdep()
gov_term ← execute.getgov()
 /* Find governor index and dependent index.*/
dep_index ← execute.getdepindex()
gov_index ← execute.getgovindex()
/* Check condition and terminate loop.*/
    if(relationship = nn or relationship = amod) then
/* Store all the dependent terms which have the same governor term in a String
variable and then append the Governor terms in the end of the last dependent term.*/
      if(prev_gov_index = null or gov_index = prev_gov_index) then
            candidate_term = candidate_term + dep_term
            prev_gov_index ← gov_index
            prev_gov_term ← gov_term
continue
      else
            candidate_term = candidate_term + prev_gov_term
candidate_term_vector = candidate_term_vector + candidate_term
end if
else if(relationship = prep_of) then
/* insert 'of' after dependent and before governor */
candidate_term = gov_term + 'of' + dep_term
             candidate_term_vector = candidate_term_vector + candidate_term
else if(relationship = conj_and) then
/* Insert 'and' after the first governor and before the dependent */
            candidate_term = gov_term + 'and' + dep_term
            candidate_term_vector = candidate_term_vector + candidate_term
else if(relationship = nsubj) then
candidate_term ← dep_term
            candidate_term_vector = candidate_term_vector + candidate_term
end if
FLUSH candidate_term
end for
return candidate_term_vector
```

The first word of the output is the antecedent noun that a pronoun refers to. In the above example, Albert Einstein and (1,1) means that Albert Einstein is word number 1 in sentence number 1 (sentence number 'comma' word number). 'He (2,1)' means that 'He' is word number 1 in sentence number 2, which also refers to Albert Einstein in the first sentence. The same goes for 'He (3,1)' and so on. Antecedent nouns in a document are then replaced with each linked pronoun, as shown in Table 4.

Table 3. Extracted multi-word candidate terms in the information system domain.

One-word candidate terms	Two-word candidate terms	Three-word candidate terms	Four-word candidate terms
Database	Data types	Intelligent tutoring system	User interface management system
Hypermedia	Information browser	Database logical design	Physical component of computer system
Internet	Word processing	Database physical design	World Wide Web Consortium
Ontology	Multimedia software	Object oriented databases	Natural language processing technique
Spreadsheets	Knowledge base	Multimedia information system	Database logging and recovery
Hardware	Hypertext navigation	Education support system	Information storage and retrieval
Benchmarking	Textual data	Information system application	System and information theory
Dictionaries	Query formulation	Decision support system	General class oriented education
Video	Data mining	Online information services	Group and organization interfaces
Hypertext	Database administration	Information processing activities	Content analysis and indexing

Table 4. Extracted noun–pronoun relationship from sample sentences.

Input:
'Albert Einstein was a German theoretical physicist. He was born in 1879. He is generally considered the most influential physicist of the 20th century. He received the Nobel Prize in physics for the service of theoretical physics. In his research career, he published more than 300 scientific papers. He died in 1955.'

Output:
ALBERT EINSTEIN (1,1), HE (2,1), HE (3,1), HE (4,1), HIS (5,2), HE (5,5), HE (6,1)
Anaphor–antecedent pairs:
(1,1) Albert Einstein ← (2,1) He,
(1,1) Albert Einstein ← (3,1) He,
(1,1) Albert Einstein ← (4,1) He,
(1,1) Albert Einstein ← (5,2) his,
(4,2) his ← (5,5) He,
(4,2) his ← (6,1) He
Text excerpt with substitution:
Albert Einstein was a German theoretical physicist.
< Albert Einstein > was born in 1879.
< Albert Einstein > is generally considered the most influential physicist of the 20th century.
< Albert Einstein > received the Nobel Prize in physics for the service of theoretical physics.
In < Albert Einstein's > research career, < Albert Einstein > published more than 300 scientific papers.
< Albert Einstein > died in 1955.

3.3. Candidate relationship extraction

Candidate terms (concepts) in a concept map are semantically linked through the candidate relationships (link) and can make meaningful propositions. We extract the semantic relations from a text document to determine the linguistic relationships among the candidate terms. This module is critical to the system's overall performance. Extracting the candidate relationship consists of two phases: (1) taxonomic relationship extraction; and (2) non-taxonomic relationship extraction.

3.3.1. Taxonomic relationship extraction. When interpreting a text document for the construction of a concept map, it is important to consider the taxonomic relations that form useful propositions. A simple method for extracting a taxonomic relationship is the well-known lexico-syntactic patterns called Hearst patterns [30]. The main idea is that, if a term NP1 is an instance or a hyponym of the term NP0, then there is a taxonomic relationship between NP1 and NP0. A total of seven lexico-syntactic patterns are used in our proposed system for the extraction of taxonomic relationships as listed in Table 5. Table 6 shows example sentences to apply the listed patterns in the information system domain.

Taxonomic relations can also be extracted for concept map construction by applying a useful heuristic defined in the CRCTOL system for ontology learning [31] based on string matching algorithms. The heuristic states that, for terms of the form [word, head], if there is a [head] in the text document, establish a taxonomic relationship between [word, head]

Table 5. Hearst patterns for taxonomic relationship extraction.

Pattern no.	Pattern definition	Relationship
Pattern 1	NP0 such as {NP1, NP2, …,(and | or)} NPn	Hyponym(NPi, NP0)
Pattern 2	Such NP0 as {NP, } * {(or | and)} NP	Hyponym(NP, NP0)
Pattern 3	NP {, NP}* {,} or other NP0	Hyponym(NP, NP0)
Pattern 4	NP {, NP}* {,} and other NP0	Hyponym(NP, NP0)
Pattern 5	NP0 {,} including {NP, } * {or | and} NP	Hyponym(NP, NP0)
Pattern 6	NP0 {,} especially {NP, } * {or | and} NP	Hyponym(NP, NP0)
Pattern 7	NP1 is a kind of | type of NP0	Hyponym(NP1, NP0)

Table 6. Extracted taxonomic relationships from sample sentences using Hearst patterns.

Pattern no.	Sample sentences with extracted taxonomic relations between the terms
Pattern 1	Typically, ontologies can be generated from various data types such as textual data, dictionaries, knowledge bases, semistructured schemata and relational schemata. (Textual Data, Data Types), (Dictionaries, Data Types), (Knowledge bases, Data Types), (Semistructured Schemata, Data Types), (Relational Schemata, Data Types)
Pattern 2	… searches by such search engines as Google, Yahoo, and MSN. (Google, Search Engine), (Yahoo, Search Engine), (MSN, Search Engine)
Pattern 3	Internet Explorer, Google Chrome, Mozilla or other Web Browsers … (Internet Explorer, Web Browser), (Google Chrome, Web Browser), (Mozilla, Web Browser)
Pattern 4	… storage devices, software, speakers and other accessories. (Storage Devices, Accessories), (Software, Accessories), (Speakers, Accessories)
Pattern 5	Apple products, including Mac, iPhone, iPad, and iPod … (Mac, Apple product), (iPhone, Apple product), (iPad, Apple product), (iPod, Apple product)
Pattern 6	… most Relational database management systems, especially Oracle, SQL Server, and MySQL. (Oracle, Relational Database Management System), (SQL Server, Relational Database Management System), (MySQL, Relational Database Management System)
Pattern 7	GATE is a type of NLP tool that extracts typed dependencies from a text using syntactic and discourse knowledge. (GATE, NLP Tool)

and [head]. For instance, the word 'Database' has a taxonomic relationship with 'Image Database' and 'Statistical Database' because both the candidate terms have the same [head] as the candidate terms, which is the head of these terms.

3.3.2. Non–taxonomic relationship extraction. To find the non-taxonomic relationships between the candidate terms to form meaningful propositions, we identify and extract these relationships by exploiting the dependency grammar. Verbs and auxiliaries are the best candidates in text documents for non-taxonomic relationships. The terms *aux* (auxiliary), *auxpass* (passive auxiliary), *nsubj* (nominal subject), *nsubjpass* (passive nominal subject), *xcomp* (open clausal complement) and *rcmod* (relative clause modifier) are the type dependencies in the Stanford Parser that extract the relationships between candidate terms. We use the same procedure to extract the candidate verb as used for candidate term extraction in

Section 3.1. For instance, the sentence 'Ontologies have been developed to capture the knowledge of a real world domain' is parsed and the typed dependencies are generated using the Stanford Parser as shown below:

```
nsubjpass(developed-4, Ontologies-1)
aux(developed-4, have-2)
auxpass(developed-4, been-3)
root(ROOT-0, developed-4)
aux(capture-6, to-5)
xcomp(developed-4, capture-6)
det(knowledge-8, the-7)
dobj(capture-6, knowledge-8)
prep(knowledge-8, of-9)
det(domain-13, a-10)
amod(domain-13, real-11)
nn(domain-13, world-12)
pobj(of-9, domain-13)
```

The defined patterns output the relationship 'developed to capture' from the above sentence. At this stage, many high-frequency candidate verbs must be identified that do not signify the semantic relationships between the candidate terms (concepts) of a domain. To find the important verbs as candidate relations, we use the $VF * ICF$ metric defined by Punuru and Chen [32] and shown in equation (5), which resembles the $TF * IDF$ metric used in information retrieval.

$$VF \times ICF(V) = (1 + \log VF(V)) \times \log\left(\frac{|C|}{CF(V)}\right) \tag{5}$$

Equation (5) shows that verbs occurring with only a few sets of candidate terms are more significant, while verbs occurring with too many candidate terms tend to be overly general and do not denote important semantic relations. In equation (5), $|C|$ is the total number of candidate terms in a document, $Vf(V)$ is the count of the occurrence of verb V in a document and $CF(V)$ is the count of the candidate terms with which the verb V is associated. If verb V and candidate term C occur in the same sentence, then they are considered to be associated with each other. Table 7 shows the sample of the top extracted relations with the corresponding $VF * ICF$ values in the Information Systems domain.

3.4. Similarity measurement between candidate terms

As discussed in Section 2.2, the clustering performance of affinity propagation depends on a similarity measure between the extracted candidate terms. In order to give specific and effective similarity values to the affinity propagation algorithm for the construction of useful clusters of candidate terms, we use a novel formula proposed by Chen et al. [17]. It is based on the relationship (taxonomic and non-taxonomic) between the candidate terms in a sentence of a document and calculates the relation strength (degree of correlation) between two candidate terms as follows:

Table 7. Extracted verbs with high *VF* * *ICF* values in the Information System domain.

Extracted verb (*V*)	*VF* * *ICF* (*V*) value
Develop	29.259
Define	28.653
Include	25.012
Extract	24.741
Represent	24.236
Generate	22.611
Describe	22.543
Process	22.081
Improve	22.047
Create	21.987

$$sim_{RS}\left(x_i, x_j\right) = \log_{10}\left(\frac{n_{ij}/\max(n)}{avg\_d_{ij}^2/\max(avg\_d^2)}\right), i \neq j \tag{6}$$

where $sim_{RS}\left(x_i, x_j\right)$ is the degree of correlation between the candidate terms x_i and x_j. The variable n_{ij} is the number of times candidate terms x_i and x_j appear in the same sentence. $avg\_d_{ij}^2$ is the sum of the proximity distance of two candidate terms divided by the number of times the two candidate terms appear in the same sentence. The equation is as follows:

$$avg\_d_{ij}^2 = \frac{\sum_{m=1}^{n_{ij}} d_m^2}{n_{ij}}, i \neq j \tag{7}$$

If the '*relation strength*' of two candidate terms is higher than that of the others, this implies that their relationship is also stronger than that of the others [17]. The measured similarity values between the candidate terms are stored in a square matrix used by the AP algorithm to cluster the relevant terms on the basis of their relationships.

3.5. Clustering related candidate terms using AP

Based on the similarity measure described in Section 3.4, the clustering process performs a sequence of steps to cluster the candidate terms on the basis of their relationships (taxonomic and non taxonomic) with each other. Algorithm 2 displays the steps to show the methodology of the clustering process. To simplify, we explain Algorithm 2 in the following steps:

Algorithm 2 Clustering candidate terms on the basis of relationships between them using affinity propagation

Input: Set of N candidate terms from Data Set D ($D \in$ Terms extracted as mentioned in Section 3.1 using Stanford Parser)
Output: Clusters made of candidate terms based on the relationship between them.
/* Compute sentence level similarity between candidate Terms $X_i \in D$ and $X_j \varepsilon \in D$. The similarity is stored in Sim_RS matrix */
1: Sim_RS[][] $\leftarrow$ execute.sim$_{RS}$(X_i, X_j) //given in equation (6)
/* Assign Self Similarity value to candidate terms from Data Set D as 1 */
2: $X_i \leftarrow \{1/ \; < X_i \varepsilon D > $ where $i\ \forall$ candidate terms in $D\}$
/* Initialize messages in Responsibility Matrix and assign 0 (zero) to Availability Matrix between two candidate terms for all candidate terms in D */
3: avail[i][j] $\leftarrow$ 0
4: resp[i][j] $\leftarrow$ Sim_RS[i][j] – max$_{j' \neq j}$ {avail[i][j'], sim_RS[i][j']}
/* Repeat until no changes in Exemplars and Clusters */
/* Calculate Responsibility and Availability Message Matrices using equation (1) and (2) */
5: avail[i][j] $\leftarrow$ computeAvailability()
6: resp[i][j] $\leftarrow$ computeResponsibility()
/* Calculate Self Availability for each candidate terms in D using equation (3) */
7: selfavail[j][j] $\leftarrow$ computeSelfAvailability()
/* Select appropriate Exemplars by adding both Responsibility and Availability Matrices and messages are updated using damping factor. */
8: execute.exemplar[i][j] $\leftarrow$ resp[i][j] + avail[i][j]
9: resp[i][j] $\leftarrow$ resp[i][j] + λ
10: avail[i][j] $\leftarrow$ avail[i][j] + λ
/* Exit if Clusters are not updated. */
11: exit(0)

Input. After selection of a text document for a particular domain, candidate terms are extracted with the help of typed dependencies patterns using the Stanford Parser and data set D with N ($N > 0$) extracted candidate terms that are initialized (data set D also contains multi-word terms).

(1) Computing the $sim_{RS}(x_i, x_j)$ between candidate terms x_i and x_j using (6), where $x_i \in D$ and $x_j \in D$.
(2) Computing the self-similarities (preferences) for each candidate term in D. $x_i \in D$, $x_j \in D$, and $i \neq j$.
(3) Initializing the matrices of responsibility and availability messages as

$$a(x_i, x_j) = 0 \text{ where } x_i \in D,\ x_j \in D \text{ and } i \neq j$$

$$r(x_i, x_j) = sim(x_i, x_j) - \max_{j \neq j'} \{ sim(x_i, x_{j'}) \}$$

(4) Computing both the matrices of responsibility and availability messages using (1) and (2), respectively.
(5) Self-availability computation: computing self-availability $a(j, j)$ for each candidate term in D using (3).
(6) Adding both the above matrices to find the exemplar x_j for each candidate term (data point) x_i that maximizes the value of $r(x_i, x_j) + a(x_i, x_j)$. All messages of both matrices are then updated by applying the damping factor λ where $\lambda \in [0, 1]$.
(7) Iterate steps (4)–(6) until the result of the cluster selection stays constant for a number of iterations.

3.6. Assignment of relationship between clustered terms

After clustering the candidate terms based on the relationships between them in a text document, the final step is to assign labels for non-taxonomic relationships between the candidate terms to form valid and meaningful propositions. Here we use all the candidate verbs, extracted in Section 3.3.2, as candidate relationship labels. We use a method called Subject–Verb–Object (SVO) developed by Punuru and Chen [32]. In the SVO method, a candidate term pair (X_i, X_j) is considered as the initial term pair if both X_i and X_j are domain-specific candidate terms that occur together in at least one sentence. Also, the pair must satisfy that X_i occurs as the subject and X_j as an object in a sentence. Punuru and Chen [32] used the MINIPAR [33] shallow parser to determine the subject and object(s) of a sentence. We only consider all those initial term pairs that exist in the same cluster, because all the clusters are made on the basis of the structural relationships between the candidate terms. Therefore, the candidate terms in different clusters have no relationships with each other and cannot produce valid and accurate propositions. Once the initial candidate term pairs are obtained using the SVO method, we assign a verb with high *VF * ICF* scores, occurring together with the concept pair in a sentence.

To summarize, we start with the extraction of the candidate terms from a text document using typed dependency rules. Second, we resolve the pronominal anaphora problem by using the RAP algorithm and associate candidate terms with their pronouns. We then extract all the taxonomic and non-taxonomic relationships between the candidate terms. After extracting the relationships, we make clusters of the candidate terms using the AP algorithm based on relationships between those terms. Finally, we assign relationship labels to the candidate term pairs to make valid propositions.

4. Experimental results and discussion

Intensive use of concept maps in many fields requires well-established evaluation methods. Considering our implementation, this section investigates the performance of the generated concept map using a certain number of measures. In this study, we conducted a search for documents related to the information system domain using online literature (http://ieeexplore.ieee.org/, http://portal.acm.org/ and http://www.sciencedirect.com). A total of 65 sample documents were selected from 2007 to 2011 from the information system domain and verified by three experts to set the baseline for the construction of concept maps from these documents. All experts have doctoral degrees in the Information System field and are currently university professors. The experts annotated all the documents manually for three weeks and identified 210 qualified sentences containing 370 candidate terms, 207 non-taxonomic relations and 81 taxonomic relations. In this section, we evaluate our system for the three major tasks of candidate terms extraction, anaphora resolution for pronouns and clustering candidate terms (concepts) using AP separately. We then estimate the system's overall accuracy by evaluating the quality of the developed concept maps in the information system domain.

Based on our methodology, a prototype has been implemented using the Java programming language in an eclipse development environment. We used the Stanford JavaNLP API to extract the typed dependencies of the qualified sentences. We then implemented the proposed algorithm described in Section 3.1 to extract candidate terms and verbs from these sentences with the help of typed dependencies. We also used RAP-RDF API to implement the anaphoric resolution module in the prototype. We used the MATLAB function 'apcluster.m' to provide the structural similarity values and the extracted candidate terms in the form of a square matrix as the input to the function to generate the clusters.

4.1. Evaluating the extraction of the candidate terms from documents

The developed system utilizes the syntactic information and well-defined rules (patterns) to extract the candidate terms for the construction of a concept map. We used the Stanford Parser to extract the typed dependencies for the selected sentences. Experiments were conducted to evaluate our system performance for single and multiword candidate terms extraction. To evaluate the performance, we calculated the precision and recall using the following formulas. In addition to those individual measures, the *F*-measure provides the weighted harmonic mean of precision and recall to summarize the overall performance of the extraction process.

$$\text{Recall} = \frac{\text{valid system's derived terms}}{\text{expert derived terms}} \tag{8}$$

$$\text{Precision} = \frac{\text{valid system's derived terms}}{\text{all system derived terms}} \tag{9}$$

$$F - \text{Measure} = \frac{(2 \times \text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \tag{10}$$

As described above, the experts extracted 370 valid candidate terms from the qualified sentences. However, our system extracted 509 candidate terms, 139 of which were not pertinent; that is, the overall precision was 72%. Out of the 139 irrelevant candidate terms, 80 were single-word candidate terms, which reduced the overall precision of the system. For recall, all 370 expert-derived candidate terms were matched with the candidate terms extracted by our system. Therefore, overall recall was 100% and the *F*-measure was 84%. Results were then compared with the C/NC-value method [34]. The C/NC-value method achieved 84% precision and 87% recall in the information system domain, which represents higher precision but lower recall than our system.

4.2. Evaluating the anaphoric resolution for pronouns using RAP

Anaphoric resolution based on the RAP algorithm was applied in our system to increase proposition extraction recall by identifying the missing propositions in the qualified sentences selected by experts. The results presented in Table 8 show that more than half of the pronouns were correctly matched with their candidate terms. Table 8 also shows that the RAP algorithm had greater accuracy on intrasentential pronouns over intersentential pronouns. Introducing anaphoric resolution for pronouns in our systems increased recall by 9% by identifying the missing proposition in the sentences.

4.3. Evaluating the extracted clusters of candidate terms using AP

Candidate term clustering is an important component of the construction of a concept. In the proposed methodology, we clustered the candidate terms using AP by extracting their structural relationship with each other in the documents. The extracted clusters of the candidate terms enabled us to find the taxonomic and non-taxonomic relationships between the candidate terms that lie in the same cluster, and that reduces the execution cost by limiting the number of irrelevant searches and not allowing the relationships between all the extracted candidate terms to be found. To construct the concept map, we need the maximum number of clusters and each cluster may contain the minimum number of candidate terms because the candidate terms generally have more intrasentential relationships than intersentential relationships. Each sentence contains an average of four candidate terms. Therefore, we increased the preference value of AP to obtain the maximum number of clusters with the minimum number of candidate terms. We compared AP with the *K*-means clustering-based, hierarchical clustering-based and spectral clustering-based algorithms.

Table 8. Results of anaphoric resolution for pronouns using the RAP algorithm.

	Total	Intrasentential pronoun occurrences	Intersentential pronoun occurrences
Number of pronoun occurrences	270	190	80
Number of cases that RAP algorithm resolved correctly	155 (57% accuracy)	117 (61% accuracy)	38 (47% accuracy)

Table 9. *F*-Measure comparison of affinity propagation with other clustering-based algorithms.

	K-means	Spectral	Hierarchical	Affinity propagation
F-Measure	0.491	0.6221	0.5824	**0.6933**
Execution time (s)	5.712	1.56593	2.02902	**1.2566**

Table 10. Constructed concept map assessment in the information system domain.

Assessment question	Scale point given by experts		
	Expert 1	Expert 2	Expert 3
1. Context of the constructed concept map	7.5	7.5	7.5
2. Validity of the extracted propositions	8	7.5	8
3. Overall impact and validity of the constructed concept map	8	7.5	8

Table 9 shows a comparison of *F*-measure and execution time for the four algorithms. The table shows that AP outperformed the other clustering-based algorithms for *F*-measure and execution time for the information system domain. However, some clusters contain wrong candidate terms that have no relationship with the other candidate terms available in the same cluster, which reduces the *F*-measure. However, the *F*-measure can be increased by providing a more accurate similarity value between the candidate terms.

4.4. Evaluating the quality of extracted concept map and propositions

Since we do not yet have a gold standard concept map for the information system domain, experts' opinion are used to evaluate the constructed concept map from qualified sentences. We used a 10-point Likert scale questionnaire [35], and asked three questions of experts to rank it for the assessment of constructed concept map. We also invited three information system domain experts to fill out the questionnaire. Chen et al. [17] used the same procedure to evaluate their constructed concept map, but with different criteria for the questionnaires. The three questions in the questionnaire are given below:

1. Does the context covered in the constructed concept map align with the corresponding qualified sentences?
2. Are propositions extracted in the concept map in accordance with the expert's knowledge about the information system domain?
3. What is the overall impact and validity of the constructed concept map in the information system domain?

Table 10 shows that the confidence of experts regarding the context alignment between the constructed concept map and extracted qualified sentence was 75% on the basis of their professional knowledge, whereas they showed 78% confidence in the validity of the extracted propositions and the overall impact and validity of the concept map. Table 11 shows two constructed sub-concept maps from the qualified sentences that received high confidence value from all three experts. For the first sentence given in Table 11, 'data types' is an exemplar candidate term in the cluster after the execution of the AP algorithm. All other candidate terms of sentence 1 are associated with the exemplar owing to their high structural similarity. All of the extracted taxonomic and non-taxonomic relationships for particular sentences are then assigned and labelled in the cluster, which is considered a sub-concept map. Also, in the second sentence of Table 11, 'physical component of computer system' is an exemplar candidate term in the cluster after the execution of the AP algorithm. Some of the sample sentences with their corresponding sub-concept maps are given in the Appendix.

As mentioned in Kowata et al. [13], high-quality concept map building approaches evaluated the systems with a subjective class. In a subjective class, human experts conduct the evaluations using their own criteria to validate a concept map [13]. However, concerning the quality of concept maps, we also compared and evaluated our methodology with four approaches by defining the criteria shown in Table 12. The last two columns of Table 12 show the uniqueness of our proposed system. In brief, we use the RAP algorithm to handle anaphoric resolution, which improves the recall result by 9% to extract the propositions in the information system domain. However, recall improvement for proposition extraction depends on the applied algorithm for anaphora resolution. The last column of Table 12 shows that our system provides

Table 11. Constructed sub-concept maps with their corresponding qualified sentences

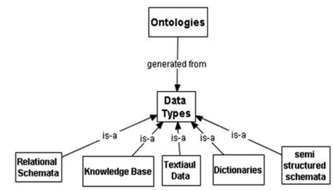
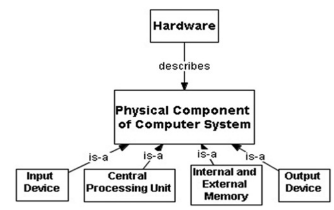
Sentence no.	Sentence	Corresponding concept map
1	Ontologies can be generated from various data types such as textual data, dictionaries, knowledge bases, semi-structured schemata, and relational schemata	
2	Hardware describes the physical component of computer system which can be categorized as an input device, a central processing unit, internal or external memory, or output device	

Table 12. Comparison of the proposed system with other approaches.

References	Domain dependency	Method	Construction methodology	Evaluation	Anaphoric resolution coverage	Context coverage
Chen et al. [17]	Domain-specific	Statistical	Automatic	Subjective	No coverage	No coverage
Valerio and Leake [20]	Domain-independent	hybrid	Manual	Subjective	No coverage	No coverage
Tseng et al. [18]	Domain-specific	Data retrieval	Automatic	Subjective	No coverage	No coverage
Zouaq and Nkambou [10]	Domain-independent	hybrid	Automatic	Subjective	No coverage	No coverage
Proposed system	Domain-independent	Hybrid	Semi-automatic	Hybrid	Coverage using RAP	Coverage using SVO

more accurate context lying between candidate terms in propositions using the SVO method. Finally, our system outperforms other approaches by applying the AP algorithm to improve computational time by assigning relationships between the candidate terms within a cluster. It helps to search the relationships between the candidate terms within a cluster rather than to compare a candidate term with all of the extracted candidate terms. Doing so reduces the computational time proportionally with the number of clusters.

5. Conclusions

In this paper, we first proposed an algorithm for automatic candidate term extraction from text documents to construct a concept map. We then introduced anaphoric resolution for pronouns to increase recall by finding more propositions in text documents. Several techniques, such as natural language processing, clustering, information retrieval and structural similarity measuring, were applied to achieve high performance. Our main contributions are as follows:

(1) our proposed system provides a rule-base of syntactic patterns to extract even multi-word candidate terms and relationships from documents.
(2) We introduced anaphoric resolution for pronouns for the first time for the construction of concept maps. This increased the recall by 9% by finding more propositions associated with pronouns.
(3) The clustering module automatically clustered the relevant candidate terms with respect to a text document for a particular domain and found the taxonomic and non-taxonomic relationships between the candidate terms within a cluster.

This helps to minimize the execution time by not requiring consideration of all the extracted candidate terms for a particular relationship. Domain experts verified that the constructed concept maps are in high accordance with their knowledge and perceptions of the information system domain. However, there are also some limitations to our proposed system. First, some extracted candidate terms were not related to the domain of interest owing to the limitation of typed dependencies of the parser, which reduces the overall precision. Second, if the structural similarity values of the different pairs of the candidate terms are 0 or unavailable, then they are grouped in one cluster even if they are irrelevant to each other in a document. We plan to enrich the pattern rule base with a new structure and further explore ways of expressing patterns for the extraction of candidate terms and relations. We also plan to introduce a new and effective algorithm for anaphoric resolution for pronouns to replace the RAP algorithm.

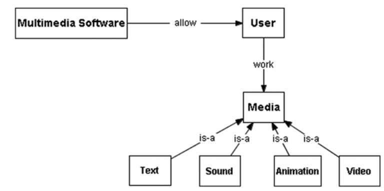
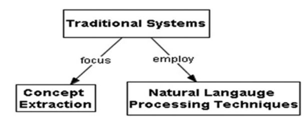
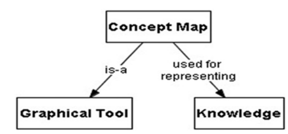
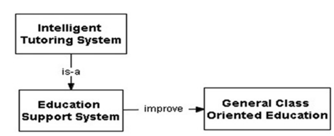
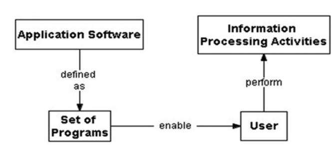
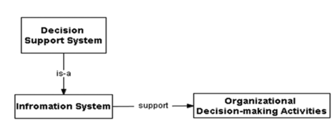
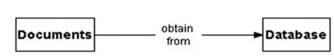
References

[1] Novak JD and Cañas AJ. The theory underlying concept maps and how to construct and use them. Florida Institute for Human and Machine Cognition, 2008.
[2] Villalon J and Calvo RA. Concept maps as cognitive visualizations of writing assignments. *Journal of Educational Technology and Society* 2011; 14(3): 6–27.
[3] Gouli E, Gogoulou A, Papanikolaou K and Grigoriadou M. COMPASS: An adaptive web-based concept map assessment tool. *Proceedings of the first international conference on concept mapping*, Pamplona, 2004, pp. 295–302.
[4] Chang SN. Externalising students' mental models through concept maps. *Journal of Biological Education* 2007; 41(3): 107–112.
[5] Van Boxtel C, Van der Linden J and Kanselaar G. Collaborative learning tasks and the elaboration of conceptual knowledge. *Learning and Instruction* 2000; 10(4): 311–330.
[6] Kinchin IM, Hay DB and Adams A. How a qualitative approach to concept map analysis can be used to aid learning by illustrating patterns of conceptual development. *Educational Research* 2000; 42(1): 43–57.
[7] Willerman M and MacHarg RA. The concept map as an advance organizer. *Journal of Research in Science Teaching* 2006; 28(8): 705–711.
[8] Okebukola PA. Can good concept mappers be good problem solvers in science? *Research in Science and Technological Education* 1992; 10(2): 153–170.
[9] Lloyd CV. The elaboration of concepts in three biology textbooks: Facilitating student learning. *Journal of Research in Science Teaching* 1990; 27(10): 1019–1032.
[10] Zouaq A and Nkambou R. Building domain ontologies from text for educational purposes. *IEEE Transactions on Learning Technologies* 2008; 1(1): 49–62.
[11] Villalon JJ and Calvo RA (eds). Concept map mining: A definition and a framework for its evaluation. In: *Proceedings of the IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, WI-IAT*, 2008, pp. 357–360.
[12] Gómez-Pérez A and Manzano-Macho D. An overview of methods and tools for ontology learning from texts. *The Knowledge Engineering Review* 2004; 19(3): 187–212.
[13] Kowata J, Cury D and Boeres M. A review of semi-automatic approaches to build concept maps. In: *Proceedings of the fourth international conference on concept mapping*, Via del Mar, Chile, 2010, pp. 120–127.
[14] Zouaq A and Nkambou R. Evaluating the generation of domain ontologies in the knowledge puzzle project. *IEEE Transactions on Knowledge and Data Engineering* 2009; 21(11): 1559–1572.
[15] Lappin S and Leass HJ. An algorithm for pronominal anaphora resolution. *Computational Linguistics* 1994; 20(4): 535–561.
[16] Frey BJ and Dueck D. Clustering by passing messages between data points. *Science* 2007; 315(5814): 972–976.
[17] Chen N-S, Kinshuk, Wei C-W and Chen H-J. Mining e-learning domain concept map from academic articles. *Computers and Education* 2008; 50(3): 1009–1021.
[18] Tseng SS, Sue PC, Su JM, Weng JF and Tsai WN. A new approach for constructing the concept map. *Computers and Education* 2007; 49(3): 691–707.
[19] Clariana R and Koul R. A computer-based approach for translating text into concept map-like representations. In: *Proceedings of the first international conference on concept mapping*, 2004, pp. 296–303.
[20] Valerio A and Leake D. Jump-starting concept map construction with knowledge extracted from documents. In: *Proceedings of the second international conference on concept mapping*, 2006, pp. 296–303.
[21] Sue PC, Weng JF, Su JM and Tseng SS. A new approach for constructing the concept map. In: *Proceedings of IEEE International Conference on Advanced Learning Technologies*, 2004, pp. 76–80.
[22] Hagiwara M. Self-organizing concept maps. In: *Proceedings of IEEE international conference on systems, man and cybernetics: intelligent systems for the 21st century*, 1995, Vol. 1, pp. 447–451.
[23] Cooper JW. Visualization of relational text information for biomedical knowledge discovery. In: *Proceedings of ACM SIGIR workshop: information visualization interfaces for retrieval and analysis*, 2003.
[24] Lau RYK, Song D, Li Y, Cheung TCH and Hao JX. Toward a fuzzy domain ontology extraction method for adaptive e-learning. *IEEE Transactions on Knowledge and Data Engineering*, 2009; 21(6): 800–813.

[25] Matykiewicz P, Duch W and Pestian J. Nonambiguous concept mapping in medical domain. In: *Proceedings of Artificial Intelligence and Soft Computing–ICAISC*, 2006, pp. 941–950.
[26] Covington MA (ed.). A fundamental algorithm for dependency parsing. In: *Proceedings of the 39th annual ACM southeast conference*, 2001, pp. 95–102.
[27] Stevenson M and Greenwood MA (eds). Comparing information extraction pattern models. In: *Proceedings of the workshop on information extraction – beyond the document*. Stroudsburg, PA: Association for Computational Linguistics, 2006, pp. 12–19.
[28] Klein D and Manning CD (eds). Accurate unlexicalized parsing. In: *Proceedings of the 41st annual meeting on association for computational linguistics*, Vol. 1, 2003. Stroudsburg, PA: Association for Computational Linguistics, 2003, pp. 423–430.
[29] McCord M. *Slot grammar. Natural language and logic*. Lecture Notes in Computer Science, Vol. 459. Berlin: Springer, 1990, pp. 118–145.
[30] Hearst MA. Automatic acquisition of hyponyms from large text corpora. In: *Proceedings of the 14th conference on computational linguistics*, Vol. 2. Stroudsburg, PA: Association for Computational Linguistics, 1992, pp. 539–545.
[31] Jiang X and Tan AH. CRCTOL: A semantic-based domain ontology learning system. *Journal of the American Society for Information Science and Technology* 2009; 61(1): 150–168.
[32] Punuru J and Chen J. Learning non-taxonomical semantic relations from domain texts. *Journal of Intelligent Information Systems* 2012; 38(1): 191–207.
[33] Lin D. MINIPAR: A minimalist parser. *Maryland Linguistics Colloquium, University of Maryland, College Park, MD*, 1999.
[34] Frantzi K, Ananiadou S and Mima H. Automatic recognition of multi-word terms: The C-value/NC-value method. *International Journal on Digital Libraries* 2000; 3(2): 115–130.
[35] Jacoby J and Matell MS. Three-point Likert scales are good enough. *Journal of Marketing Research* 1971; 8(4): 495–500.

Appendix

Table A1. Constructed sub-concept maps with their corresponding qualified sentences

<table>
<tr><th>Sentence no.</th><th>Sentence</th><th>Corresponding concept map</th></tr>
<tr><td>1</td><td>Multimedia software allows users to work with media such as text, sound, animation and video.</td><td>
</td></tr>
<tr><td>2</td><td>Traditional systems employ natural language processing techniques and focus only on concept extraction.</td><td>
</td></tr>
<tr><td>3</td><td>A concept map is a graphical tool used for representing knowledge.</td><td>
</td></tr>
<tr><td>4</td><td>Intelligent tutoring systems are educational support systems, which improve general class-oriented education.</td><td>
</td></tr>
<tr><td>5</td><td>Application software can be defined as a set of programs that enable users to perform information processing activities.</td><td>
</td></tr>
<tr><td>6</td><td>Decision support system is an information system that supports organizational decision-making activities.</td><td>
</td></tr>
<tr><td>7</td><td>The documents may be obtained from the database.</td><td>
</td></tr>
</table>