PATENT DOCUMENT SUMMARIZATION USING CONCEPTUAL GRAPHS

Pattabhi R K Rao and Sobha Lalitha Devi

AU-KBC Research Centre, MIT Campus Anna University, Chromepet, Chennai, India

ABSTRACT

In this paper a methodology to mine the concepts from documents and use these concepts to generate an objective summary of the claims section of the patent documents is proposed. Conceptual Graph (CG) formalism as proposed by Sowa (Sowa 1984) is used in this work for representing the concepts and their relationships. Automatic identification of concepts and conceptual relations from text documents is a challenging task. In this work the focus is on the analysis of the patent documents, mainly on the claim's section (Claim) of the documents. There are several complexities in the writing style of these documents as they are technical as well as legal. It is observed that the general in-depth parsers available in the open domain fail to parse the 'claims section' sentences in patent documents. The failure of in-depth parsers has motivated us, to develop methodology to extract CGs using other resources. Thus in the present work shallow parsing, NER and machine learning technique for extracting concepts and conceptual relationships from sentences in the claim section of patent documents is used. Thus, this paper discusses i) Generation of CG, a semantic network and ii) Generation of abstractive summary of the claims section of the patent. The aim is to generate a summary which is 30% of the whole claim section. Here we use Restricted Boltzmann Machines (RBMs), a deep learning technique for automatically extracting CGs. We have tested our methodology using a corpus of 5000 patent documents from electronics domain. The results obtained are encouraging and is comparable with the state of the art systems.

KEYWORDS

Concept Mining, Document Summarization, Abstractive Summary, Conceptual Graphs, Machine Learning, Restricted Boltzmann Machines, Patent Documents

1. INTRODUCTION

Concept Mining is the task of extracting the concepts found in a text document. Concept is a representation of an idea or entity and it can be the smallest unit, the word or a chunk which is the phrase. Identification of concepts from text documents requires natural language processing and machine learning. Concept identification is a non-trivial task and helps in understanding the texts, given a semantic representation of the text. In the literature most commonly used methods to identify concepts have been through the use of thesaurus such as WordNet, dictionaries or lexicons. Here in this work machine learning technique is used for identifying concepts and their relationships.

In the present study, patent documents are chosen because of their complex sentence constructions, both at syntactic level and semantic level. Patent documents are technical and legal documents which contain technology innovations written in legal language style. A patent document consists of several sections such as abstract, prior art, novelty or claim, the methodology and figures.

In general, a legal document comprises of long and semantically very complex sentences. When this is combined with technical writing the sentences become highly complex and are difficult to

DOI: 10.5121/ijnlc.2017.6302

analyse. It has been observed that a single sentence in a patent document consists of more than 200 words. Hence a patent document contains sentences that are syntactically and semantically complex, which are very difficult to be analysed even by human beings.

Today there is great need for the automatic analysis of patent documents for applications such as prior art search, patent infringement etc. Prior art search of patents require summaries of the claims section. There is a need to generate abstractive summaries of the patent documents. Various methods are used for summarization of a text, but are not giving a good summary. It is found that concept based summarization will be more semantically driven and give cohesion to the summary generated. "A Summarizer is a system whose goal is to produce a condensed representation of the content of its input for human consumption" (Mani, 2001). Most of the methods used for automated summary generation have the end result as a collection of sentences which do not have connectivity of topic or it can be said as the cohesion of the text is not present. Here in this work, it is aimed to bring in this cohesion to the summary through the conceptual graph based summarization.

Automated summarization is an important area of research in NLP. One of the popularly known earliest works on text summarization is by Luhn (Luhn, 1958). He proposed that frequency of a word in articles provide a useful measure of its significance. Significance factor was derived at sentence level and top ranking sentences were selected to form the auto abstract.

A variety of automated summarization schemes have been proposed in the last decade. (Mihalcea, 2004; Mihalcea and Tarau, 2004; Mihalcea et al, 2004; Gupta and Siddiqui, 2012) have been proposed for single document summary generation. NeATS (Lin and Hovy, 2002) is a sentence position, term frequency and term clustering based approach. MEAD (Radev et al., 2004) is a centroid based approach. Iterative graph based Ranking algorithms, such as Google's Page- Rank (Brin and Page, 1998), Kleinberg's HITS algorithm (Kleinberg, 1999) are used in social networks, web-link analysis and more recently in text processing applications.

An algorithm which forms a semantic network of all the sentences in the document is proposed in this paper. And from this semantic network, abstractive summary of the document is formed. The semantic network is generated using conceptual graph (CG).

The major contributions of this work are as follows:

i. Using CGs a semantic knowledge representation formalism, concepts are extracted.
ii. Extended the definition Sowa (Sowa, 1984), (discuss in detail what constitutes a concept and how concepts are formed in section 2 of this paper).
iii. Concepts and their relationships are mined and developed as CG by automated means. Most of the earlier works in literature have used partial automation for the development of CGs.
iv. The formalism of CGs helps in generating an abstractive summary. Most of the earlier works are extractive summaries. And those that have generated abstractive summary use rule based approach, which raises the issue of scalability and adaptability for various genres.
v. CGs are scalable and could be adopted for any language. Though here in this work it is demonstrated using English, but the same could be used for any language from any language family.

This paper is further organized as follows. In the next sub-section 1.1, a brief description of the background of conceptual graphs is given for the benefit of readers. A more detailed description can be found in (Sowa 1984). Section 2, describes our modification and extension of concept definitions used to facilitate our work. The methodology and the approach for automatic

conceptual graph extraction are described in Section 3. In Section 4, we describe the methodology for summary generation. And section 5 concludes the paper.

1.1. Related Works

A conceptual graph (CG) is a graph representation of logic based on the semantic networks of artificial intelligence and existential graphs of Charles Sanders Peirce. John Sowa states the purpose of conceptual graphs as "to express meaning in a form that is logically precise, human readable and computationally tractable" (Sowa, 1984). Mathematically, a CG is a bipartite, directed, finite graph; each node in the graph is either a concept node or relation node. Concept node represents entities, attributes, states, and events, and relation node shows how the concepts are interconnected. A node (concept or relation) has two associated values: a type and a referent or marker; a referent can be either the single generic referent, or an individual referent. Thus a CG consists of a set of concept types and a set of relation types. CGs were first introduced by John Sowa in his work on database interfaces (Sowa, 1976). It illustrated CGs for representing natural language questions and mapping them to *conceptual schemata*. Each schema contained a declarative CG with attached *actor nodes* that represented functions or database relations. Sowa (1984) explains CGs for the representation of natural language texts. The concept nodes represent entities, attributes, events, actions. And relation nodes represent the kind of relationship between the two concept nodes.

The main advantage of representing natural language text in the form of CG is that, CGs can be easily converted to any Knowledge Interchange Format such as first order logic, hence semantic processing is possible. The challenge in automatic representation of a natural language text in CG is the identification of concepts and the relationships between them. Hence automatic identification of concepts and conceptual relations is very important for the purpose of semantic representation and inference.

Conceptual graphs have been used in applications such as question answering systems and information retrieval systems to improve the performance of the systems. Molla and Van (2005) build "AnswerFinder" - a framework for QA systems – in TREC 2004. Here in the graph patterns between the questions and answers is learnt. The conceptual graphs are based on translation of logical forms of sentences in the training data of question and answers given in TREC 2004. The graph matching algorithm is based on the maximal graph overlap. Here they obtain average accuracy of 21.44% and average mean reciprocal ratio of 25.97%.

Siddiqui and Tiwary (2006) use CGs for representing text for the information retrieval task. They use CG in conjunction with VSM model for representation. Here the information retrieval task is done in a two phased manner. In the first phase the relevant documents are retrieved using the VSM model. The resultant documents are used as input for the CG model and the finally most relevant documents are retrieved. Here a small set of semantic relations are used to construct CGs, these relations are developed based on the syntactic patterns. CACM-3024 data collection is used for the experiments. They show an increase of 34.8% in precision and overall 7.37% improvement in retrieval performance.

Montes-y-Gomez et al (2000, 2001) discuss about information retrieval using CGs. In this work they present methodology for comparison of two conceptual graphs. The similarity measure is based on the dice coefficient, takes into consideration both concepts and relations of the graph while calculating similarity.

Conceptual graphs are also used in developing knowledge base. Karalopoulos et al., (2004) use CGs for representing geographic knowledge. In their work, they create a CG for each

geographical definition. All similarly created CGs are inter-connected to form a network, thus a geographic knowledge base is developed.

A conceptual graph is represented mainly in two forms viz., i) Display form and ii) Linear form. The display form uses the traditional graph form, where concept nodes are represented by rectangular boxes and relation nodes are represented by ovals. In the linear form concepts are represented by square brackets and relation nodes are represented using parenthesis. To represent these graphs internally inside the computer system we use a list data structure consisting of triplet value (c1, c2, r), where c1 is concept one and c2 is concept two and r is the relationship between the concepts c1 and c2. This triplet structure can be again represented using traditional matrix representation which is currently followed by information systems. The example below would give more insight into CGs.

Example 1: English Sentence: "Marie hit the piggy bank with a hammer."

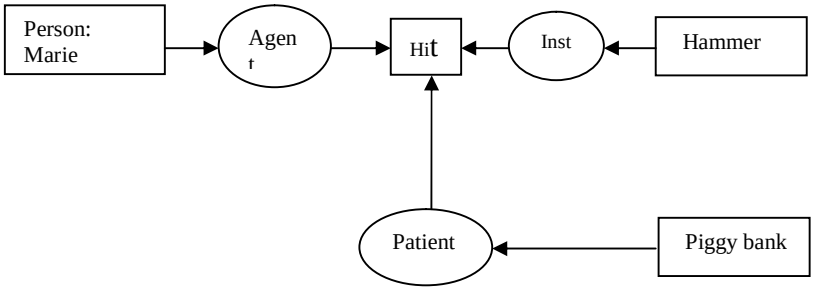

Figure 1. Conceptual Graph – Display form – Example Sentence 1

The figure 1, shows the display form of conceptual graph for the example 1 sentence. The concepts are "Marie", "Hit", "Hammer" and "the piggy bank". And these concepts are connected by the relationships "agent", "instrument" and "patient" respectively. From the graph we can infer the following: the subject "Marie" had "hit" the object "piggy bank" using the instrument "hammer".

2. OUR DEFINITION OF CONCEPTS AND CONCEPT FORMATION

One of the most important parts of this work is the identification of concepts in a document. Thus it is necessary for us to understand what a concept is. In general a concept is defined as a representation or expression of a thought or idea conceived in a human mind. This can be a perception of an object, a natural phenomenon, or a feeling experienced by us. Smith and Medin (1981), in their study of concepts have summarized three views or approaches on definition of concepts viz.

a) Classical Approach
b) Probabilistic Approach
c) Prototype Approach

Classical approach is one of the most popularly used in the formal treatments in Mathematics and logic. Sowa in his [Sowa, 1984] works on CG has followed the classical approach. He defines concepts in terms of percepts. Percepts are the units of perception. He states, "the process of perception generates a structure 'u' called a CG in response to some external entity or scene 'e'. For every percept 'p' there is a concept 'c', called the interpretation of 'p'." Though he describes

about abstraction, the emphasis is on the objects and perception. He describes words for the concepts. Sowa considers percepts as the basic unit for the concept formation (Sowa, 1984). His work elaborates on the structural part of the graph formalism. He does not specifically describe what constitutes a concept and how the different concepts are formed. Here we explore on what constitutes a concept from the semantic and computational perspective.

In our analysis of the documents it is observed that the words in isolation have a particular meaning and have a different meaning when they are in collocation with immediate words. Also the meaning of individual words in a phrase varies with the meaning of the phrase. Thus it can be concluded that the Phrases and words in collocation are to be considered as a single unit. We term these phrases or collocation words which have unique meaning in a particular context as 'a Semantic Unit'. And we consider the semantic unit as the basis for our concept definition. Thus we define, for every semantic unit 'SU', there is a concept 'c' which is directly related to the semantics or meaning of 'SU'. In the document look for phrases and collocations of words having unique meaning. By this modifying the basic unit for concept we have substantially modified and extended the definition of concept given by Sowa to facilitate our work.

What constitute a semantic unit is discussed here. The syntactic and semantic tags are used for defining the semantic units. The grammatical categories which form semantic units are described below:

i) Multi Word Expressions:

Multiword expressions (MWEs) are expressions which are made up of at least two words and which can be syntactically and/or semantically idiosyncratic in nature. Moreover, they act as a single unit. MWEs are idiosyncratic interpretations that cross word boundaries.
Examples: 'took off', 'by and large', 'take off', 'in short' (Frozen forms)

ii) Endocentric phrases:

An endocentric phrase consists of two words, in which one is the head and other is modifier and both together would specify or narrow down the meaning of the head. This is a special case of multi word expression.

Examples: 'diesel motor', 'house boat'

iii) Exocentric phrases:

An exocentric phrase consists of two words whose meaning is different from the constituent words. This is also a special case of MWE. Examples: 'pale face', 'white collar', 'pick pocket'.

iv) Possessive Noun phrases:

Possessive noun phrases show the ownership or possession of an object or person. These phrases consist of two entities. The first entity owns or possesses the second entity. Examples: 'cattle's pasture', 'John's book'

v) Noun phrases:

These are set of words which together form a noun have one meaning and would refer to a single entity. Examples: 'smart phone', 'running water', '

vi) Verb phrases:

These are set of words which together form a verb and have one meaning and would refer to a single action, activity. Examples: 'mild boiling', 'fast bowling'. In the above it has been discussed on how a single concept is formed. Further it gives in detail how these are formed with examples.

Concept Formation:

Here we describe how two or more words would combine to form a new concept. A new concept 'c3' would be formed by the combination of concepts c1 and c2:

If
- *concept c1 modifies c2 i.e., c1 is the modifier of c2*
- *and c2 is specified by c1 i.e., c2 is the specifier of c1*

There are different types of combination of words which are formed by the grammatical features associated with the words in concept such as specifier, modifier and multi word expression. The explanation given below shows how such combination can happen.

• The new concept c3 is a kind or type of c1 or c2. In general the type of c3 is similar to the type of c2 since c2 forms the head of the combination.

Example [c3] – Thematic [c1] + Connection [c2]
Example [c3] – Mobile [c1] + Phone [c2]

• The new concept c3 is a specialization of c2 and has different meaning not obtained from c1 and c2.

Example [c3] – Love [c1] + Life [c2],
[c3] – deep [c1] + fry [c2],
[c3] – continuous [c1] + production [c2]

Here we observe that for concepts the most likely part-of-speech (POS) categories of lexical words involved in the formation are noun-noun, adjective-noun, adverb-noun, noun-verbal noun. The prepositions or postpositions generally do not form concepts. They indicate the relationship between the concepts.

3. CONCEPTUAL GRAPH EXTRACTION FROM PATENT DOCUMENTS

The most prominent and technically important section of a patent document is the claim or novelty section which describes and defines the claims of the invention. This describes the core novelty of the invention, for which protection is claimed by the inventors. A claim in the patent document could be classified into two types, independent and dependent claims. Independent claims introduce a unique novelty feature of the invention, whereas a dependent claim describes more about the novelty already mentioned in an independent claim. Most of the independent claims on an average consist of 250 – 400 words in a sentence. These sentences cannot be parsed correctly by the general parsers available in the open source and also crash the system. We make use of Information Extraction (IE) techniques to cull out the novelty/claim part of the patent document.

One of the similar works was done by (Yang & Soo, 2012), where the sentences in the claim section were split into different parts using few heuristics, so that those could be parsed by a parser. The output thus obtained from a parser is used for developing CG. The problem with the sentence splitting as explained in (Yang & Soo, 2012) is that it does not retain the full meaning of the sentence and there could be information loss. Problems in such simplification for ease of processing is that of gap-filling expressions, ordering of phrases in the sentences, maintaining discourse coherence will lead to improper interpretation of sentences.

Another major challenge in the construction of CGs is defining concepts and relations between the concepts. A patent document describes main invention which can be an object or a process or a product (which again is an object). Along with this it also describes the components or parts of the object and sub-processes. It also describes the properties, characteristics, uses and advantages. In general a concept is defined as an abstract idea conceived mentally by a person.

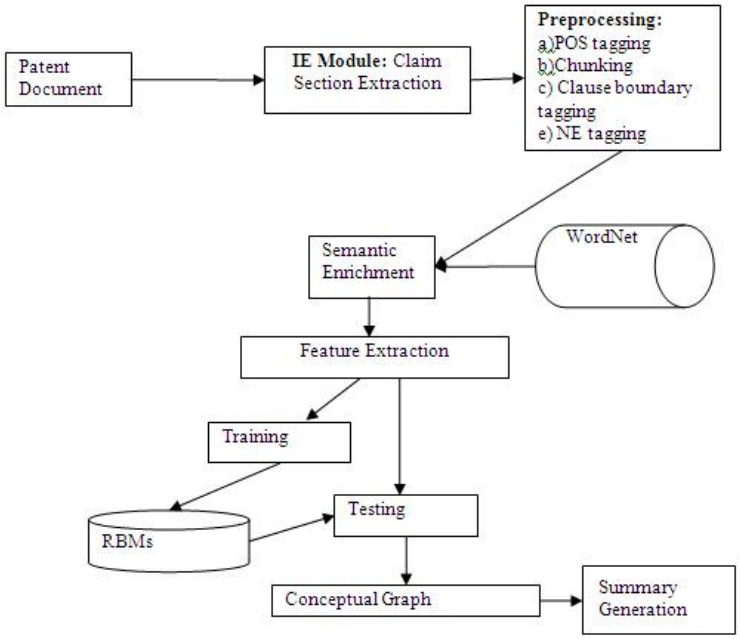

Figure 2. CG Extraction – Process Flow diagram

A concept will have same meaning across the languages, but have different lexical representations. A concept can be represented by a single word or a group of words. According to this definition every noun becomes a concept and identifying the relation between concepts is difficult. Figure 2, shows the overall system process flow diagram with focus on CG extraction.
First we cull out the novelty or claims part of the patent document. For this part we use information extraction technique. We extract the core part of patent information which are of particular interest to the end user. The information extracted are

(i) Novelty (Key aspect of the invention) (ii) Dependent/Independent Claims (Secondary Inventions related) (iii) Use (Application area of the invention) (iv) Advantage of the invention (compared to prior invention).

This module typically uses extraction patterns/rules that represent specific linguistic contexts to recognize the critical and novel information related to the above–mentioned 4 fields. The rules developed for extracting these data are also generic but a very little modification is required according to the domain.

Also to extract information about a conceptual topic, there is a need to have a dictionary of extraction patterns to recognize the relevant information about the topic. For identifying the information related to novelty, a dictionary which contains **36** extraction patterns. Likewise for identifying use there is a dictionary containing **210** extraction patterns and for advantage **423** patterns. Again linguistic rule based approach is utilized for extracting the Dependent /Independent claims. Articles and Prepositions related to the claim section of the patent is also identified and analyzed for extracting the Dependent /Independent claims.

Once these parts from the patent document are extracted, in further processing and analysis of the patent document only the claims section is used.

Each document is pre-processed to obtain syntactic and semantic information using Natural Language Processing (NLP) tools. The sentence splitter and tokenizer are done using grammar and heuristic rules. We make use of Brill's POS tagger (Eric, 1994) and fnTBL (Ngai and Florian, 2001) for Part-of-Speech tagging and chunking respectively. We have developed a named entity recognizer (Malarkodi et al., 2012) using Conditional Random Fields (CRFs), a machine learning technique (Lafferty, 2001). After the NLP processing of the documents, we do the concept identification. The next sub sections describes in detail the extraction of concepts and relations and formation of conceptual graph.

3.1 Automatic Identification of Concepts and Relations

There are two sub-modules in this component; the first one is the concept identification module and second is the relation detection module. In the concept identification module, the concepts as defined in section 2 are automatically identified using deep learning. Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations (Srivastava et al., 2013; Hinton and Salakhutdinov, 2006). Deep learning is defined as a class of machine learning algorithms that use a cascade of many layers of nonlinear processing units for feature extraction and transformation. And learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts. In this work Restricted Boltzmann Machine (RBMs) which is one of the methods in deep learning is considered. In earlier work by Pattabhi and Sobha (2015), they had described on identification of concepts and their relationships using RBMs. The same implementation is used for identifying the concepts in this work.

An RBM is a probabilistic model. It models a distribution by splitting the input space in many different ways. RBM is a type of Boltzmann Machines (BMs). BMs are a particular form of log-linear Markov Random Field (MRF), for which the energy function is linear in its free parameters. To make them powerful enough to represent complicated distributions which go from the limited parametric setting to a non-parametric one. We consider that some of the variables are never observed (they are called hidden). By having more hidden variables (also called hidden units). We can increase the modelling capacity of the Boltzmann Machine (BM). Restricted Boltzmann Machines (RBMs) further restrict BMs to those without visible-visible and hidden-hidden connections. Unlike other unsupervised learning algorithms such as clustering, RBMs discover a rich representation of the input. RBMs are shallow, two-layer neural nets. The first layer of the RBM is called the visible, or input, layer, and the second is the hidden layer. A graphical depiction of a RBM is shown in figure 3.

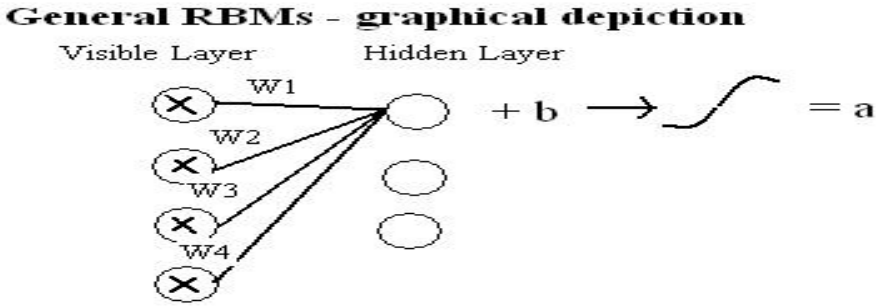

Figure 3. Graphical depiction of RBMs

Each circle in the graph above represents a neuron-like unit called a node, and nodes are simply where calculations take place. The nodes are connected to each other across layers, but no two nodes of the same layer are linked. That is, there is no intra-layer communication – this is the restriction in a restricted Boltzmann machine. Each node is a locus of computation that processes input, and begins by making stochastic decisions about whether to transmit that input or not. Let x be the value of the visible node (or input value) and w1is the weight at node 1, then the result obtained is given by the equation below:

*activation f((weight w * input x) + bias b) = output a* *(1)*

So equation (1) when expanded will become

activation f((xw1 + xw2+xw3+xw4) +b) = a (output) *(2)*

.

Because inputs from all visible nodes are being passed to all hidden nodes, an RBM can be defined as a *symmetrical bipartite graph*.

In this work, three levels of data is provided as input in the visible layer. The first level is the words or tokens. The second level is the part-of-speech (POS) information and the third level is the named entity information. The modified graphical depiction of the RBM will be as shown in figure 4.

In our case we are giving input as word, POS and NE.

So x=<y1,y2, y3> where y1= word, y2=POS and y3=NE

Thus in our case equation (1) will be as follows

*f((<y1,y2,y3>*w1+<y1,y2,y3>*w2+<y1,y2,y3>*w3+<y1,y2,y3>*w4)+b) = a* *(3)*

and when expanded (3) will become

*f(((y1w1*y2w1*y3w1)+(y1w2*y2w2*y3w2)+(y1w3*y2w3*y3w3)+(y1w4*y2w4*y3w4))+b)=a*

(4)

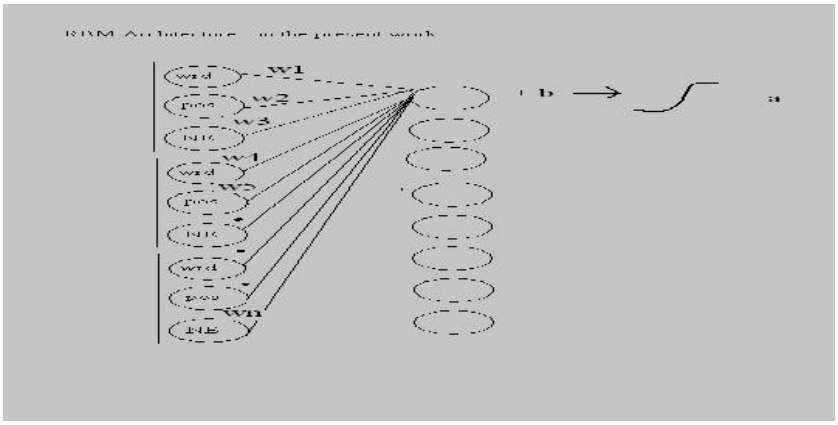

Figure 4. RBM in the Present Work - graphical depiction

The motivation behind using the word, POS and NE tags for RBMs is that the unsupervised RBMs can detect the structures in the input and automatically obtain better feature vectors for classification. Most of the earlier NLP works have only used words as input for training the RBMs. The aim of the present work is to identify concepts from the word representations. The POS tag and NE tag help to add sense and semantic information to the learning. The NE tag will help in identifying whether they are attributes of objects, phenomenon's, events etc. This gives indications on kind of concepts while learning and thus help in concept identification. RBMs is modelled as pairs of 3-ary observations. The 3-ary consists of word, POS and NE Tag.

A RBM is a generative stochastic neural network that can learn probability distribution over its set of inputs. Restricted Boltzmann machines are trained to maximize the product of probabilities assigned to training set V (a matrix, each row of which is treated as a visible vector v),

$$\underset{w}{argmax} \prod P(v)$$

or equivalently, to maximize the expected log probability of a training sample selected randomly from V.

$$\underset{w}{argmax}\, E[\log P(v)]$$

These three levels of data in the visible layer (or input layer) are converted to vectors of n-dimension and passed to the hidden layer of the RBM. The word vectors, POS vectors and NE vectors are the vector representations. These are obtained from the word2vec. These are also called as word embedding. Word embedding, in computational linguistics, referred as distributional semantic model, since the underlying semantic theory is called distributional semantics (Srivastava, 2013). The real valued n-dimensional vector for each level is formed using the word2vec algorithm. Word2vec creates or extracts features without human intervention and it includes the context of individual words/units provided in the projection layer. Word2vec is a computationally-efficient predictive model for learning word embedding's from text. The context comes in the form of multiword windows. Given enough data, usage and context, Word2vec can make highly accurate word associations. Word2vec expects a string of sentences as its input. Each sentence – that is, each array of words – is vectored and compared to other vectored lists of words in an n-dimensional vector space. Related words and/or groups of words appear next to each other in that space. The output of the Word2vec neural net is a vocabulary with a vector

attached to it, which can be fed into the next layer of the deep-learning net for classification. The DL4J Word2vec API is used for this purpose.

We have obtained optimal hyper parameters for good performance by performing several trials. The main hyper parameters which need to be tuned include choice of activation function, number of hidden units, learning rate, dropout value, and dimensionality of input units. In this 20% of training data is used for tuning these parameters. The optimal parameters include: 200 hidden units, rectilinear activation function, 200 batch size, 0.025 learning rate, 0.5 dropout and 25 training iterations. The best development set accuracy was obtained at 80 dimensional word vector and 5 dimensional POS and NE tag vectors. Thus for each word we have 3-arys word vector, POS vector and NE vector, consisting of 90 dimensions. The output layer uses softmax function for probabilistic multi-class classification. The corpus data used for learning the Word2vec embeddings to convert the data to a 90 dimension of 3-arys for input to the RBMs is describe in section 5. The RBM is trained and using the RBMs the concepts given in document are identified. Once the concepts are extracted there is need to identify the relationships between them and thus form a semantic network.

3.1.2 Relation Identification

Concepts are always interconnected and do not exist in isolation. Concepts are connected with each by various relationships. Thus there is need to identify these various relationships that exist between the concepts to form a conceptual graph which is a semantic network. The figure 5 shows the process flow diagram in the relation identification module.

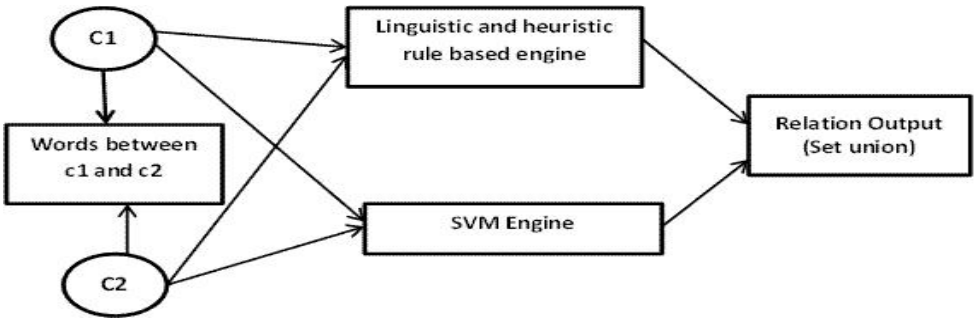

Figure 5. Relation Detection Module Process flow.

Relationship identification module uses a hybrid approach. Here there are two sub modules, rule based engine and SVM engine. The output of both engines is merged.

The linguistic rules are used initially to identify well defined relations. The linguistic rules use syntactic structure of the sentence. Some of the linguistic rules are described below:

(i) If concept c1 is a verb/verbal phrase and concept c2 is a noun/noun phrase and there are subordinators such as “after”, “later” before the c2 then these are markers of temporal relations. Using these temporal relationships one can infer senior-junior relationships, if this exists between two person concepts. For example in the sentence “John joined ABC corp after Marie”. This shows John is junior to Marie.
(ii) If concept c1 and c2 are connected by be verbs such as “is”, then there exists “is a” or “sub-type” relationship.

A preposition relation mapping table which defines different relations for each type of prepositions associated between verb-noun, noun-noun concepts is developed. Support Vector

Machine (SVM) classifier to identify relations independent of the rule based engine is used here. TinySVM toolkit is used for the SVM implementation (Taku Kudo, 2002). The output of SVM classifier and the output of the rule based engine are merged to get the set of all relations. In the SVM engine output only those relations which get higher confidence score of more than 0.75 are considered as valid relations. The features used for training SVM engine are the words and POS feature.

4. SUMMARY GENERATION

The <concept-relation-concept> tuple obtained from is a bipartite graph consisting of two classes of nodes "concepts" and "relations" and form a CG. For a sentence many such tuples are obtained depending on the number of clauses. These are merged into the sub-graphs of the sentence to form a CG. Sub-graphs are merged by computing clique-sum. In this method two graphs are merged by merging them along the shared clique. A clique in a graph is a subset of vertices in which every two vertices are connected by an edge. Each tuple can be considered as a clique. Hence the shared cliques are identified and merged to form a unified network of the conceptual graph for all the sentences. This complete CG is the semantic network of the document. This is a kind of inheritance network, where the lower nodes correspond to more specific regularities and the upper nodes to more general ones. This hierarchy allows multiple inheritances. The document summary generation has the following two steps.

(i) Formation of clusters of longest chain of nodes in the graph.
(ii) Select nodes that are in the longest chains for summary generation. If there are more than five longest chains obtained then restrict to 30% of the total number of sentences in the claims part of the patent document.
(iii) Summaries are generated by forming the template based generation approach in which concept nodes are connected by specific joiners depending on the relation type.

4.1 Algorithm: Identification of cluster of longest chain of nodes

This is similar to identification of longest path problem in a directed acyclic graph. The semantic network obtained from the earlier steps is a directed acyclic graph. The longest path problem for a general graph is not as easy as the shortest path problem because the longest path problem doesn't have optimal substructure property. In fact, the Longest Path problem is NP-Hard for a general graph. However, the longest path problem has a linear time solution for directed acyclic graphs. The idea is similar to linear time solution for shortest path in a directed acyclic graph. Here in this approach since it is dealing with bipartite graphs and is for the purpose of summary generation it is needed to identify the most significant nodes, we adopt Hopcroft-Karp algorithm to identify maximal matches of the graph. The Hopcroft-Karp algorithm (Hopcroft and Karp, 1973; Blum, 2001) which is implemented has been described below.

Let U and V be the two sets in the bipartition of G, and let the matching from U to V at any time be represented as the set M. The algorithm is run in phases. Each phase consists of the following steps.

a) A breadth-first search partitions the vertices of the graph into layers.
b) The free vertices in U are used as the starting vertices of this search and form the first layer of the partitioning.
c) At the first level of the search, there are only unmatched edges, since the free vertices in U are by definition not adjacent to any matched edges.
d) At subsequent levels of the search, the traversed edges are required to alternate between matched and unmatched. That is, when searching for successors from a vertex in U, only

unmatched edges may be traversed, while from a vertex in V only matched edges may be traversed.

e) *The search terminates at the first layer k where one or more free vertices in V are reached.*
f) *All free vertices in V at layer k are collected into a set F. That is, a vertex v is put into F if and only if it ends a shortest augmenting path.*
g) *The algorithm finds a maximal set of vertex disjoint augmenting paths of length k. This set may be computed by depth first search from F to the free vertices in U, using the breadth first layering to guide the search:*
h) *The depth first search is only allowed to follow edges that lead to an unused vertex in the previous layer, and paths in the depth first search tree must alternate between matched and unmatched edges.*
i) *Once an augmenting path is found that involves one of the vertices in F, the depth first search is continued from the next starting vertex.*
j) *Each one of the paths found in this way is used to enlarge M.*

The algorithm terminates when no more augmenting paths are found in the breadth first search part of one of the phases.

Now the sentences are selected from the documents which contain the nodes or vertices of the maximal match. A threshold is put to the number of sentences to be considered for summary. The number of such selected sentences is restricted to 10% of the total sentences in the whole set of documents.

5. EXPERIMENTS AND RESULTS

5.1 Data Set

In this work 5000 USPTO patents from electronics domain were collected. The documents pertaining to electronic gadgets were chosen. The patent documents were obtained from the USPTO, full-text web service by providing keywords/key phrases such as ‘capacitive keypad’, ‘lcd mobile display’, to the USPTO web service and were downloaded.

A set of 2000 patents were manually annotated for the concepts and relations. The tagset used for annotating <concept-relation-concept> tuple is as follows:

a) **R-relation_type-<idx>-S** -- Here <idx> is relation index number in a sentence it is from 1,2,..N. *relation_type* specifies the type name of the relation. “S” indicates start. If relation is indicated by more than one word, then B-I-O standard format of representation is used.

b) **C1-<idx>-S** -- Here C1 refers to concept 1 of Relation R<idx>. “S” indicates start.

c) **C2-<idx>-S** -- Here C2 refers to concept 2 of Relation R<idx>. “S” indicates start.

A single concept can occur inside different relations, to accommodate this we have used the notation of indexing. As it can be seen in the above tagset, <idx> is the indexing used for this purpose. A sample annotation schema for a partial sentence is shown below.

A Portable/C2-1-S Electronic/C2a-1-S {attribs}/R-attribs-1-S appliance/C1-1-S,C1-2-S comprising/R-comprise-2-S : a keypad/C2-2-S,C1-3-S having/R-has-3-S a plurality/C1-3-S of/C1-3-I keys/C1-3-I

5.2 Conceptual Graph Extraction

This section describes our experiments to evaluate our RBMs as a feature extraction method. The system results are compared with those of earlier works towards automatic identification of concepts and relations. Shih Yao (2012) work is similar to the present work. In this earlier work the data sets used were patent documents from chemistry domain. The work of Shih Yao is not exactly comparable with the present work, since the methodology used by them is completely different and also the data set differs from the current work. But still in table 1, the results of Shih Yao is presented. In the training phase, as explained in earlier section 3, the documents are pre-processed. After pre-processing the words are tagged with concept classes. The concepts are represented as vectors of 100 dimensions using the Word2Vec algorithm. These vectors are then presented in the format as required by the RBMs and trained. A 10 fold experiment is performed, where 90% for training and 10% for testing is taken. The evaluation metrics used are the precision, recall and f-measure as used in the earlier works. Table 1 shows the average of 10 fold experiment results for <concept-relation-concept> tuple. The precision is calculated as the ratio of 'correctly extracted concept-relation-concept (CRC) tuples by the system' to 'total CRC tuples extracted by the system'. Whereas Recall is the measure of coverage, thus it is calculated as the ratio of 'correctly extracted CRC tuples by the system' to 'gold CRC tuples for the document'.

Table 1. Results of the Concept relation extraction

Method	Precision (%)	Recall (%)	F-measure (%)
Shih-Yao [2012]	78.75	70.2	74.22
Pattabhi et al [2013]	73.3	68.3	70.71
Our Approach	79.34	72.54	75.79

For a sentence several such tuples depending on the number of clauses is obtained. These sub-graphs obtained for a sentence are merged to form a CG for the whole sentence. Sub-graphs are merged by computing clique-sum. In this method two graphs are merged by merging them along the shared clique. A clique in a graph is a subset of vertices in which every two vertices are connected by an edge. Each tuple can be considered as a clique. The shared cliques are identified and merged to form a single CG. For example let us consider the following patent claim sentence

(1) *"A portable electronic appliance comprising: a keypad having a plurality of keys, wherein each of the plurality of keys is arranged so as to actuate a respective mechanical switch so as to provide a first type of user input; and an impedance sensing means disposed integrally with the keypad so as to provide a second type of user input that is characterized as non-mechanical, wherein the impedance sensing means operates as a proximity sensitive touchpad, wherein the keypad and the impedance sensing means are coextensive, wherein the impedance sensing means is of a size that is always adaptable for use in a hand-held device, and wherein the impedance sensing means is disposed under the keypad."*

Here the following <concept-relation-concept> tuples are obtained as machine output. Graph operation "join" is performed to form a single conceptual graph as shown in Fig. 6.

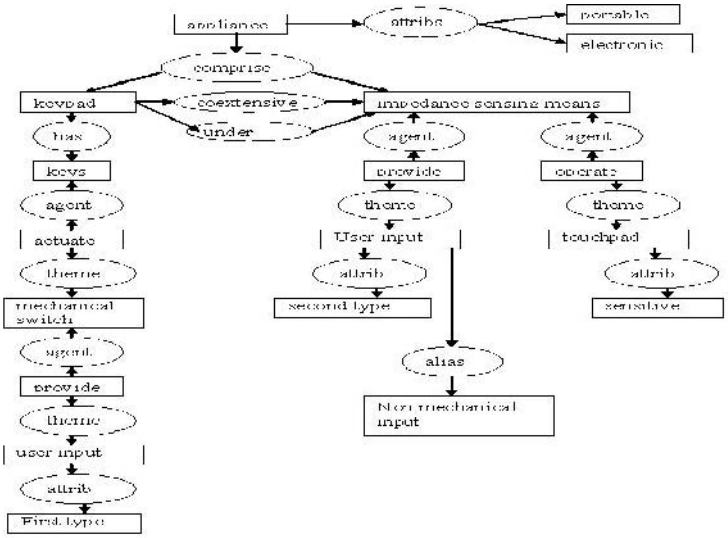

Figure 6. Conceptual Graph diagram for example (1)

The analysis of the results shows that the non-identification of concepts gave maximum error of 50%. In the output it is observed either two concepts are combined into one or only partially identified. For example in the sentence

(2) *"an actuator assembly having an elongate plunger along a central vertical axis and having at one end a plurality of co-planar actuator surfaces arrayed initially in a horizontal plane"*

The system has combined two concepts "elongate plunger" and "central vertical axis". In some instances the concept "display screen" is identified as two different concepts "display" and "screen". This is mainly due to the pre-processing errors such as NP chunking error, where many nouns are clubbed together to form a single NP or split as different NPs. As said earlier in some instances system identifies "display screen" as two different NPs.
Though the system resolves most of the verb senses properly, in 30% of the error cases, the system does not identify the sense correctly and this has led to incorrect identification of relations. For example for the word "disposed" concept mapped from the WordNet concept is "possession", whereas the correct one would be "contact".

5.3 Summary Generation

Here there is no training phase as we directly use the CGs produced in the first step. As explained in section 4 after the CGs are obtained the semantic network of the CGs obtained for all sentences in a document are formed. And from the semantic network the summary is generated. For summary evaluation, the commonly used automatic evaluation tool called, the ROUGE package, which was developed by Lin (2004) is used. ROUGE is based on the n-gram overlap between a system generated summary and a set of reference summaries. It measures a summary quality by counting overlapping units, such as the word n-gram, word sequences, and word pairs between the candidate summary and the reference summaries.

The ROUGE-N recall score is computed using the formula shown in Figure 4. Where, ROUGE-N is an n-gram recall between a system generated summary and a set of reference summaries. 'n'stands for the length of the n-gram, gram and $Count_{match}$ (gram) are the maximum number of n-grams co-occurring in a system generated summary and a set of reference summaries. The older

versions of the ROUGE package, such as Versions 1.1, 1.2, 1.2.1, and 1.4.2, only used a recall based score for summary evaluation. Whereas, the newer version of the ROUGE package--ROUGE 1.5.5--evaluates summaries based on three metrics such as ROUGE-N precision, ROUGE-N recall, and the ROUGE-N F-Score, where N can be 1, 2, 3, 4 etc. Thus, the ROUGE toolkit reports separate scores for 1, 2, 3, and 4-grams, and also for the skip bigram. The ROUGE Version 1.5.5 is used for the system evaluation. Among the various ROUGE scores, the unigram and bigram based ROUGE score (ROUGE-1 & 2) have been shown to most agree with human judgment (Lin and Hovy, 2003). The ROUGE-2 metric is found to have high correlation with human judgments at a 95 % confidence level and hence used for evaluation. We have obtained an average 0.2198 F-measure of ROGUE score. The results obtained are comparable with the state of the art. This result is comparable with the results reported by other summarizer systems in MultiLing 2015 (Giankopolus, 2015) evaluation exercise. The data used in MultiLing 2015 are not the same as used here. The documents used in that evaluation exercise are general News articles.

6. CONCLUSION

This paper has presented a system for concept mining and abstractive summarization of patent documents. This uses conceptual graph formalism for representation. One of the main advantages is that the summary is coherent and also easy to scale. This approach can be adopted for any language, with a very minimal customization, since this uses conceptual graph principles of knowledge representations. An average F-measure of 0.2198 ROUGE score is obtained which is comparable with state of the art. The main objective of the current work was to ascertain how capturing of structure of a sentence and thereby of the document would help in generating abstractive document summaries. This is very useful and got good results. The use of CGs helped in the capture of the structure and the semantics and helped in generating abstractive summary.

REFERENCES

[1] Amati, G. and Ounis, I., (2000) “Conceptual Graphs and First Order Logic”, The Computer Journal, 43(1):1-12.

[2] Athanasios, K., Margarita, K., and Marinos K., (2004) “Geographic Knowledge Representation Using Conceptual Graphs”, In the Proceedings of 7th AGILE Conference on Geographic Information Science, Greece.

[3] Blum, N., (2001) “A Simplified Realization of the Hopcroft-Karp Approach to Maximum Matching in General Graphs”, Tech. Rep. 85232-CS, Computer Science Department, Univ. of Bonn.

[4] Bezdek J.C., (1981) “Pattern Recognition with Fuzzy Objective Function Algorithms”, Plenum Press, New York.

[5] Brin and Page, L., (1998) “The anatomy of a large scale hypertextual Web search engine”, Computer Networks and ISDN Systems, Vol 30, pp. 1-7.

[6] Brill, Eric., (1994), “Some Advances in transformation Based Part of Speech Tagging”. In Proceedings of the Twelfth International Conference on Artificial Intelligence (AAAI-94), Seattle, WA.

[7] Deigo, M., Menno V. and Zaanen., (2005) “Learning of Graph Rules for Question Answering”, In the Proceedings of the Australasian Language Technology, Workshop 2005, Sydney, Australia. 15–23.

[8] Erkan and Radev, D., (2004) “Lexpagerank: Prestige in multi document text summarization”, In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain.

[9] Hinton, G. and Salakhutdinov, R., (2006) “Reducing the dimensionality of data with neural networks”, Science, 313(5786):504 – 507.

[10] George Giannakopoulos, Jeff Kubina, John M Conroy, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, Massimo Poesio., (2015), “MultiLing 2015: Multilingual Summarization of Single and Multi-Documents, On-line Fora, and Call-center Conversations”, In: Proceedings of SIGDIAL, Prague pp. 270-274.

[11] Hopcroft, J.E., Karp, R.M., (1973) "An n5/2 algorithm for maximum matchings in bipartite graphs", SIAM Journal on Computing, Vol 2(4), pp. 225-231, doi:10.1137/0202019.
[12] John F. S., (1976) "Conceptual Graphs for a Data Base Interface", IBM Journal of Research and Development 20(4). 336–357.
[13] John F.S., (1984) "Connceptual Structures – Information Processing in Mind and Machine". Addison Wesley.
[14] Lin and Hovy, E.H., (2002) "From Single to Multi document Summarization: A Proto-type System and its Evaluation", In Proceedings of ACL-2002.
[15] Lin, C.Y., (2004) "ROUGE: A package for automatic evaluation of summaries", In Proceedings of the Workshop on Text Summarization Branches Out, Barcelona, Spain.
[16] Lin, C.Y and Hovy, E., (2003) "Automatic evaluation of summaries using n-gram co-occurrence", In Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003), Edmonton, Canada.
[17] Luhn, H.P., (1958) "The automatic creation of literature abstracts", IBM Journal of Research and development, Vol 2(2), pp. 159-165.
[18] Inderjeet Mani., (2001) "Summarization Evaluation: An Overview", In Proceedings of NTCIR.
[19] Malarkodi C.S and Sobha Lalitha Devi, (2012) "A Deeper Look into Features for NE Resolution in Indian Languages", In Proceedings of Workshop on Indian Language Data: Resources and Evaluation, LREC 2012, Istanbul.
[20] McKeownand, K. and Radev, D., (1995) "Generating summaries of multiple news articles", In Proceedings of the 18th Annual International ACM, Seattle, WA, pp.74-82.
[21] Mihalcea R., (2004) "Graph-based ranking algorithms for sentence extraction, applied to text summarization", In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain.
[22] Mihalcea and Tarau.P., (2004) "TextRank - bringing order into texts", In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain.
[23] Mihalcea, P.T., and Figa, E., (2004) "PageRank on semantic networks, with applica-tion to word sense disambiguation", In Proceedings of the 20th International Con-ference on Computational Linguistics (COLING 2004), Geneva, Switzerland.
[24] Lafferty, J., McCallum, A., and Pereira, F., (2001) "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", In Proceedings of the 18th International Conference on Machine Learning (ICML-2001), pp.282-289.
[25] Menaka S., Pattabhi R.,K.,R. and Sobha, L.D., (2011) "Automatic Identification of Cause-Effect Relations in Tamil Using CRFs", In A. Gelbukh (eds), Springer LNCS volume 6608/2011, 316-327.
[26] Taku Kudo., (2002) "TinySVM Tool Kit: http://chasen.org/~taku/software/TinySVM".
[27] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean., (2013) "Efficient Estimation of Word Representations in Vector Space", In Proceedings of Workshop at ICLR.
[28] Miller, G.A., (1995) "WordNet: A Lexical Database", In Comm.of ACM. 38(11). 39-41.
[29] Mineau, G., Missaoui, R. and Godinx, R., (2000) "Conceptual modeling for data and knowledge management", In Data & Knowledge Engineering, (33)137-168.
[30] Montes-y-Gomez, M., Lopez-Lopez, A. and Gelbukh, A., (2000) "Information Retrieval with Conceptual Graph Matching", In LNCS Volume. 1873, Springer-Verlag.
[31] Montes-y-Gomez, M., Gelbukh, A., Lopez-Lopez, A., and Baeza-Yates, R., (2001) "Flexible Comparison of Conceptual Graphs", In LNCS, Volume 2113, Springer-Verlag.
[32] Ngai, G., and Florian, R., (2001) "Transformation-Based Learning in the Fast Lane", In Proceedings of the NAACL'2001, Pittsburgh, PA, 40-47.
[33] Pattabhi RK Rao, Sobha Lalitha Devi and Paolo Rosso., (2013) "Automatic Identification of Concepts and Conceptual relations from Patents Using Machine Learning Methods", In Proceedings of 10th International Conference on Natural Language Processing (ICON 2013), Noida, India.
[34] Tanveer J. S., Umashanker, Tiwary., (2006) "A Hybrid Model to Improve Relevance in Document Retrieval", Journal of Digital Information Management, Vol. 4(1). 73-81.
[35] Trevor, Cohn, Philip, Blunsom, (2005) "Semantic Role Labeling with Conditional Random Fields", In the Proceedings of CoNLL.
[36] Shih-Yao Y., Von-Wun, S., (2012) "Extract Conceptual Graphs from Plain Texts in Patent Claims", Journal of Engineering Applications of Artificial Intelligence. 25(4). 874-887.
[37] Srivastava, N., Salakhutdinov, R. R. and Hinton, G. E., (2013) "Modeling Documents with a Deep Boltzmann Machine", In Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI).

AUTHORS

Dr. Sobha Lalitha Devi is a Scientist/Faculty at AU-KBC Research Centre, Anna University Chennai. Her research interests are Anaphora resolution, Natural language processing. She has versatile experience in Language technologies.

Pattabhi RK Rao is a Research Scholar at AU-KBC Research Centre, Anna University Chennai. His research interests are Information retrieval, Conceptual graph representation.