ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 8(6), 536–547, March 2015* DOI : 10.17485/ijst/2015/v8i6/61070

Ontology Based Concept Hierarchy Extraction of Web Data

K. Karthikeyan[1*] and V. Karthikeyani[2]

[1]Research and Development Centre, Bharathiar University, Coimbatore, Tamil Nadu, India; kk_karthikeyan2007@rediffmail.com
[2]Department of Computer Science, Thiruvalluvar Government Arts College, Rasipuram, Tamil Nadu; India; drvkarthikeyani@gmail.com

Abstract

This paper proposes the method of Ontology Based Concept Hierarchy Extraction of Web Data. This helps to extract Concept Hierarchy efficient way for ontology construction. It is very useful for learning the ontology from the text in more efficient way. In General, Natural Language is Complexity and Uncertainty. The existing system used either Statistical based learning or logic based learning Techniques. Statistical based learning techniques gives solution only for complexity and Logic based techniques gives solution for uncertainty alone. But the Statistical Relational Learning Techniques give solution for both Complexity and Uncertainty. So, our proposed system uses Statistical Relational Learning Technique, named Markov Logic Network. Markov Logic Network is a technique in which identify the concept in the domain and order the candidate terms in hierarchical way. An experimental result provides the best concept hierarchy extractions compared to the state-of-art methods.

Keywords: Concept Hierarchy Extraction, Hearst Pattern, Markov Logic Network, Ontology, Semantic Web.

1. Introduction

Ontology is an explicit formal specification of a shared conceptualization of a domain of interest, where formal implies that the ontology should be machine readable and shared that it is accepted by a group or community. Ontology play a crucial role in many net and net related applications as they are suggests that by these can be model and share info throughout a selected domain. And also, Ontology is a collection of concepts. Ontology Engineering is a methodology to construct the Ontology. There are two ways to construct the Ontology, one is Ontology Editor and another one is Ontology Learning. Ontology Editor is application software in which to construct the ontology manually, but, Ontology Learning is a semi automated method to learn Ontology from text.

Ontology learning layer cake is contributing to a better understanding of the OL tasks[3]. This ontology learning layer cake can be used to classify an OL approach according to the task that it aims at.

Generally, Ontology Learning methodology has the following task, term extraction, concept extraction, concept hierarchy extraction, semantic relation extraction, axiom learning and ontology population.

Term Extraction consists individual terms, concept is knowledge about the object, concept hierarchy establish relationship among the concepts, semantic relation is also establish non taxonomic relation among the concept, axiom learning is learning rules in the domain and finally, constructed the ontology is known as ontology population.

The proposed system is move from concept extraction to concept hierarchy. Originally, Ontology learning system is deal with natural language process domain. The NLP domain is always uncertainty and complexity[26]. The Probability framework is suitable to give correct solution to uncertainty problem only and the same time, logical based framework is very suitable for complexity problem alone. Many techniques have been proposed to extract Concept Hierarchy based on Statistical analysis and

*Author for correspondence

Linguistic[31]. But they are not able to remove the noise in the text and also they are not able to extract relationship between the terms. But Statistical Relational Learning methodology, like Markov Logic Network gives solution for both of the problem in single framework. So far there is no learning methodology used MLN to give the solution for NLP problem.

Here, the system uses Markov Logic Network to identify concept in domain and extract the hierarchy among the term of concepts to form tree structure. The system gives best result to compare art of the state of methodology.

To afford the good results of concept hierarchy extraction, we use PREHE (Probabilistic Relational Hierarchy Extraction). PREHE[23] is a technique for extracting concept hierarchies from natural language. This could be used in probabilistic relational learning. This technique[24] is mainly used for link prediction. The goal of link prediction[2] is to determine whether a relation exists between two objects of interest from the properties of those objects and possibly other known relations. The formulation of this problem in MLNs is identical to that of collective classification, with the only difference that the goal is now to infer the value of R(xi; xj) for all object pairs of interest, instead of C(x; v). Formal Concept Analysis is used in order to extract concepts by grouping terms related to the same set of verbs. The hierarchy is extracted by applying the partial ordering operator to the extracted concepts.

Our work is also related with some existing effort of web structure of Markov Logic Network. Our goal is to improve the extraction process based on hierarchical manner. To employ our process, before performing the hierarchy extraction we do some progress of pre-processing the words, concept identification. Pre-processing is used for preparing the exact meaning and syntactic meaning of a word. To find out the weight of a particular sentence or word perform the concept identification. Next, perform the task of concept labeling[25]. These methods are helpful for maintaining the collaboratively web data structural richness. As a final point, there are some works adopt to improve the concept extraction and integrating the web data of retrieval process using MLN and FCA. Link prediction process enlarges the hierarchy extraction. Here, MLN is deal with link prediction and provide structural richness of web data named as an Ontology Based Concept Hierarchy Extraction of Web Data (OBCHED) that performs the efficient results.

The rest of this paper is organized as follows: Section 2 reviews the related work. Section 3 contains the process of General Markov Logic Network details. Section 4 formulates the proposed framework of Ontology Based Concept Hierarchy Extraction of Web Data. In further section it will be called as OBCHED, gives results of the concept hierarchy extractions. Section 5 has the experimental results for the projects. Section 6 formulates the Conclusion for the projects and Section 7 contains the Future enhancement of the projects. Thus the paper clearly explains the concept of the OBCHED.

2. Markov Logic Network

The Markov Logic Networks provide a strong probabilistic modeling framework based on First-Order Logic. The statistical relative learning combines the communicatory power of data representation formalisms with probabilistic learning approaches, so facultative one to represent syntactical dependencies between words and capturing applied mathematics information of words in text. Here, the MLN process should be combined with the Markov Random Fields. The weights placed in a MLN process it may be either positive or negative. The MLN process has two kinds of constraints. There are hard constraints and soft constraints. The set of possible worlds are placed in hard constraints and also the set of impossible worlds could be placed in soft constraints.

The Markov Logic Network[15] having two types of inference tasks. There are Maximum a Posteriori (MAP) and probability inference. The aim of MAP inference is to find the most probable state of the world given some evidence. According to the truth assignments we have to maximize the sum of weights in the network. There are two approaches for learning the weights of a given set of formulas: generative and discriminative learning. Generative learning aims at maximizing the joint likelihood of all predicates while discriminative, at maximizing the conditional likelihood of the query predicates given the evidence ones. Probabilistic logical thinking aims at determinative the likelihood of a formula given a set of constraints and may be alternative formulas as proof. The likelihood of formula is that add of the possibilities of the worlds wherever it holds.

Normally, Markov Logic Networks consists of a weighted first order formule is also called as clauses or rules. In that we have to describe the truth associated weight. The probability of the particular truth assignment

can take the variable of x, its probability can be described as follows,

$$P(X=x)=\frac{1}{z}\exp\left(\sum_{f_i\in\tau} w_i \sum_{g\in Gf_i} g(x)\right)$$
$$=\frac{1}{z}\exp\left(\sum_{f_i\in\tau} w_i n_i(x)\right) \quad (1)$$

Where, g(x) is 1 means value of g has to be satisfied or otherwise not satisfied with every values in the process.

For learning[15,16,26–30] markov logic networks, it consists of two tasks. There are structure learning and weight learning. The weight learning is the independent component that can be learn weights for clauses written by a human expert. In weight learning we have to use two types of learning approaches named as generative learning and discriminative learning. The structure learning can be performed using an algorithm. The process behind the structure learning based on search methods. Beam search or shortest first search can be used to develop the candidate clauses. All candidate clauses are evaluated and added into the markov logic network.

The Alchemy software[31] is mainly used to learning the weights. Using this kind of method we have to produce the modification of finding inference. Here, we can use the exact probabilistic method of learning weights and produce the good results of inference. In that the markov blanket of a query atom can only contains the evidence atoms. The conditional patterns of this method can be described as,

$$\log P(Y=y\mid X=x)=\log\prod_{j=1}^{n} P(Y_j=y_j\mid X=x)$$
$$=\sum_{j=1}^{n}\log P(Y_j=y_j\mid X \quad (2)$$

Where, X is the set of evidence atoms and Y is the set of query atoms.

This process can help us to reduce the size of markov blanket, when the clauses are satisfied by the evidence. Exact inference is very fast because the MLN contains thousands of clauses.

The MLN weights can be derived from more relational databases. MLN weights can be calculated by the log likelihood manner.

$$\frac{\partial}{\partial w_i}\log P_w(X=x)=n_i(x)-\sum_{x'} P_w(X=x')n_i(x') \quad (3)$$

Here, the sum is overall possible databases x′ and compute the probability by using current weight vector ω.

MLN can formulate their features into social networks, language processing and spatial statistics. To optimize those process use pseudo-likelihood described as,

$$P_w^*(X=x)=\prod_{t=1}^{n} P_w(X_t=x_t\mid MB_x(X_t)) \quad (4)$$

Where, $MB_x(X_t)$ is the state of the markov blanket of a data. For first order logic we have to compute the probability. In first order logic attributes have one variable for each pair (a, b), where a is an argument of the query predicate and b is the argument of the query predicates with some same values of each pair. Each and every set of predicates have the truth assignment values. According to those values we construct the network inference. Learning weight process could be done by using the network inference values computed with the help of log likelihood method and their possible probabilities.

The MLN[30] process could be trained using a Gaussian weight prior with zero mean and unit variance. The MLN process could be used in[32] Concept Extraction for Ontology learning. The inference can be computed as quite fast. The statistical relational learning weights can be formulated in the MLN. The task performed in this processes are link prediction, link based clustering social network modeling and object identification.

3. OBCHED (Ontology Based Concept Hierarchy Extraction of Web Data)

OBCHED is an Ontology Based Concept Hierarchy Extraction of Web Data. The OBCHED describes the process of concept hierarchy extraction. Concept hierarchy is the process that contains sub process of concept identification and concept extraction. The most existing technique of hierarchy extraction was developed from the formal concept analysis and markov logic networks. In our proposed system concept hierarchy extraction based on link prediction and web data retrieval. Link prediction is used to predict the terms in learning process. The OBCHED techniques have some types. There are Pre-Processing, Concept Identification and Concept Hierarchy Extraction. The OBCHED process clearly explains in Figure 3.

3.1 Pre-Processing

GATE tool was used for Pre-processing the text corpus, the following steps involved such as: i) Tokenization, ii) POS tagging, iii) chunking and Syntactic analysis, iv) Calculating Term Weight and v) Hypernym Extraction.

i) Tokenization is the main part of learning process, because every word in the sentence named as tokens. To enlarge the keywords, the result of this portion displays the tokens of each and every sentence. In most of the application, we were using POS tag to get grammatical tags in the sentence or word corpus. The way of getting input as corpus and transform the input into a model that can be processed computationally.

ii) POS tagging leads to a) stop-word removal, b) stemming and c) lemmatization. a) Stop-Word Removal processes for removing unwanted letters in each and every word in text corpus. Here, we had to remove the letters like "is", "was", "and", "that" etc. In stop-word removal, taxonomic relationship based removal is done. b) In stemming, the stemming algorithm used to retrieves the stem of a word by removing its longest possible ending which matches one on a list stored in the English Dictionary like Word Net. Next, handles "spelling exceptions," mostly instances in which the "same" stem varies slightly in spelling according to what suffixes originally followed it. Stemming process should be remove the letters placed in the words are "ed", "ly", "ing". Usually English words constitute some morphological paradigm to assigning the lemmas. c) Lemmatization progress may be of grouping the words that belongs to the same inflectional paradigm and assigning to each paradigm its corresponding canonical form called lemma. Lemmatization process performs four steps. There are

- Removal of suffix of length
- Addition of new lemma suffix
- Removal of prefix of length
- Addition of new lemma prefix

After performing the part of speech tagging the GATE tool do the process of chunking. These processes are shown in Figure 1.

The results came from the lemmatization process having tokens, grammatical tags and lemmas.

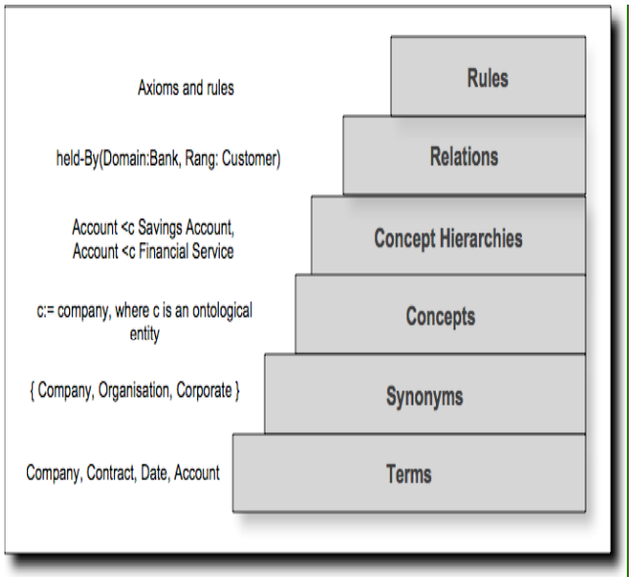

Figure 1. Ontology Learning Layer Cake.

iii) Here after for our contribution we have to do the operation of chunking. The reason for performing this operation is to highlight the set of forming words and according to these words co-ordinate the integration of web data. In chunking, set of words could be formed and display in highlight manner. GATE tool performed this operation in our implementation side. Then every part in the ontology learning progress wants to know the syntactic meaning of the words. For this purpose we examine the syntactic unit of every word. This method is very useful but is not always easy to manipulate. The options for modification provide another way to identify the categories that are relevant for both word formation (morphology) and phrase formation (syntax).

iv) The next stage of pre-processing is to calculate the term weight and also hypernym extraction. In term weighting option we have to find out the weight of every word. Term weight which is calculated by the scores of TF-IDF calculation, that is the Term Frequency and the inverse document frequency.

v) Finally, we perform the process of hypernym extraction. Hypernym extraction performed with the help of hearst pattern. To determine the possible hypernym of particular noun we use the same parsed text. Then construct the vector of each hypernym. It would be useful for identify terms made up of multiple words rather that just using the head nouns of the noun phrases.

3.2 Concept Identification

Concept Identification is an important portion covered in our proposed system. Concept identification is performed by the technique of MLN. Using MLN we have to perform the process of learning weight and inference. Figure 2 describes the process of concept identification.

For performing the learning weight we have to use the method of MLN. To find the weights in a database we have to use the Maximum a Posteriori (MAP) weight method. This means the weights that maximize the product of their prior probability and the data likelihood. Pseudo-likelihood is that the product of the conditional chance of every variable given the values of its neighbors within the data. Whereas economical for learning, it will offer poor results once long chains of inference are needed at enlarging time.

Pseudo-likelihood is systematically outperformed by discriminative coaching, it minimizes the negative conditional probability of the question predicates given the evidence ones. This learning weight can be performed by four methods. First, progress based on voted Perceptron. Here, using gradient descent algorithm use the gradient named as g, scaling based learning rate η, and to update the weight vector w, it can be represented by,

$$W_{t+1} = W_t - \eta g \tag{5}$$

The spin off of the negative Conditional Log-Likelihood (CLL) with relevancy a weight is that the distinction of the expected range of true groundings of the corresponding clause and therefore the actual range in step with the information.

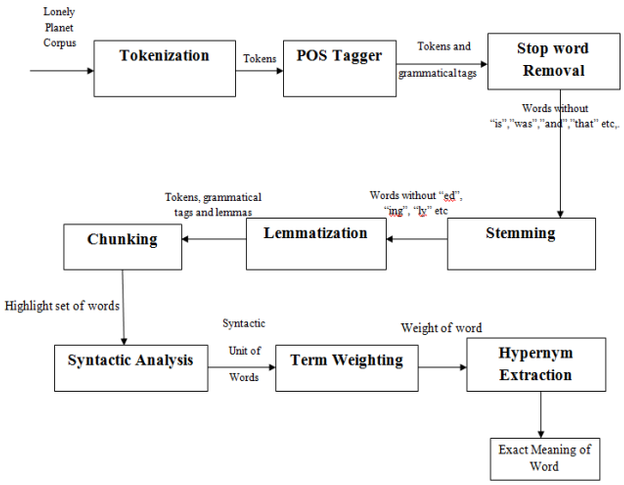

Figure 2. Pre-Processing.

$$\frac{\partial}{\partial w_i} - \log P\left(Y = y \mid X = x\right) = E_w\left[n_i\right] = n_i \tag{6}$$

Where y is the state of the non-evidence atoms in the data, and x is the state of the evidence.

The second process is the contrastive divergence. In contrastive divergence we use MCMC algorithm. The MCMC algorithmic program usually used with contrastive divergence is Josiah Willard gibbs sampling, expect for MLNs a lot of quicker various method MC-SAT is offered. Because ordered sample in MC-SAT square measure a lot of less related to than ordered sweeps in Josiah Willard gibbs sampling, they carry additional data and square measure doubtless to yield a better descent direction. Specially, the various samples square measure doubtless to be from completely different modes, reducing the error and potential instability related to choosing one mode.

The third progress is per-weight learning rates. To modify each algorithms are to own a distinct learning rate for each weight. Since standardization of each learning rate individually is impractical, we use an easy heuristic to assign a learning rate to every weight.

$$\eta_i = \frac{\eta}{n_i} \tag{7}$$

Where η is the user-specified global learning rate and n_i is the number of true groundings of the i^{th} formula. These values are being fixed, so it cannot be contribute to the variance.

The final process in the series is Diagonal Newton. In diagonal newton we just multiplying the gradient, g, by the inverse Hessian, H inverse.

$$w_{t+1} = w_t - H^{-1} g \tag{8}$$

In Diagonal Newton (DN) methodology, this uses the inverse of the diagonoized jackboot in situ of the inverse

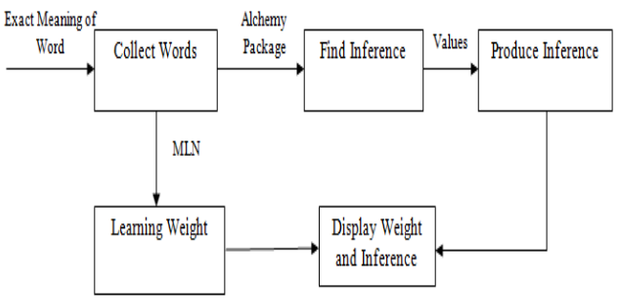

Figure 3. Concept identification.

jackboot. DN typically uses a smaller step size than the total Newton methodology. The main aim of this method is to found the step size. In each iteration, we take a step in the diagonalized Newton direction.

$$w_i = w_i - \alpha \frac{E_w[n_i] - n_i}{E_w[n_i^2] - (E_w[n_i])^2} \quad (9)$$

Then we compute the step size,

$$\alpha = \frac{-d^T g}{d^T H d + \lambda d^T d} \quad (10)$$

Where d is the search direction. For a quadratic function and $\lambda = 0$, this step size would move to the minimum function value along d.

Regarding inference we have to perform the task of finding inference using alchemy software we have to finalize the inference values of each word in the schema. Before enter into the process of concept extraction we have to know about the PLSA process. Using this method we can derive the meaningful words in the corpus. So, using PLSA we find the synonym of the word.

3.3 PLSA

PLSA (Probabilistic Latent Semantic Analysis) is normally used to capture the polysemy and synonymy in text. This was plays a role in many applications like retrieval and segmentation. To train the parameters in the PLSA we have to use one algorithm named as Expectation Maximization. This PLSA consists of three parts. There are

- Document Selection
- Probability for latent class
- Probability for words

Here, we want to discuss with the portion of web based retrieval of information using PLSA. In that process first perform the initialization of PLSA, performance of PLSA and also retrieval process. Initialization needs to train the data's in the dataset or from any other sources. It can be proceed to train good models. After trained all data's we have to collect the good accuracy of particular data. Each and every PLSA model may have the different initialization. The likelihood can play an important role in Initialization. Because, likelihood increases are to a locally optimal value with each iteration of Expectation Maximization. Using likelihood the accuracy of the corresponding model does not correlate with each other. Hoffman and brants produce the random initializations and also found the position of improve the performance. Combination of model is used to minimize the redundancies of every parameter and also minimize the expression of errors. The result derived from this approach is any one good initialization could improve the performance over simply using a number of different initializations.

Next we focus the operation of retrieval task. In information retrieval progress first, take smaller corpus, on the order of a personal document collection. Here, we have to use four document collection parts. They are MED, CRAN, CISI, CACM.

- MED (Documents related to Medicine)
- CRAN (Documents related to Institute of Technology)
- CISI (Documents related to Institute of Scientific Information)
- CACM (Documents related to Association for Computing Machinery)

For each information set, we tend to use the computed representations to estimate the similarity of every question to all the documents within the original assortment. For retrieving web based information we use the Probabilistic Latent Semantic Analysis process and get the good performance results of documents. The document collections fully based on the initialization and performance of the PLSA.

In PLSA model, we first computed the probability of each word occurring in the document, it can be represented as,

$$p(w \mid d) = \frac{p(w, d)}{p(d)} \quad (11)$$

Where, p(d) is the probability of document and p(w, d) is the probability of word and documents. Using this equation and assume that p(d) uniform to every process of documents in the corpus. This will gives a good and smooth representation of every document. Then we have to find the similarity between term distribution of the candidate document, p(w/d) and query p(w/q). These could produce the efficient way of retrieving web documents using four methods, account for the dependence on initial conditions.

The next process include in the PLSA model is to segment the text potions. Text segmentation is performing based on the similarity of probability model. Here, a text can be divided into overlapping blocks of sentences and the PLSA representation of the terms in each block, p(w/b)

is computed. The similarity between adjacent blocks b, b is computed using their probability of blocks and similarity measure. Text segmentation based on the boundaries. These boundaries can have the blocks and with close matches. According to that matches we decide the boundaries are within the k words/sentences, where k is the half of the average segment length in the test data. In order to account for the random initial values of the PLSA models, we tend to perform the entire set of experiments for each parameter setting four fold and averaged the results.

3.4 Concept Labeling

The completion of concept identification is move onto the process of concept labeling. The result came from the concept identification and term extraction can be taken into an input as concept labeling. Here, the inference and weight values can enter into the labeling process. This could be in an internal process. Our main process does not explicitly show the results of labeling. But according to the labeling results we have to move onto the next process of concept hierarchy extraction. In corpus the relation labeling can be used to retrieve the learning ontology. In this process we have to combine these S, R and also term extraction and taken into the input of hierarchy extraction.

3.5. Concept Hierarchy Extraction

In Concept Hierarchy Extraction we have to do three kinds of processes. There link prediction, hierarchy extraction using FCA and hierarchy extraction using Hearst Pattern. In that progress we have to include some additional processes. In first, link prediction task can be done with the help of MLN. The Markova logic network can extract the words in different manner. For that purpose predicts the links in an efficient manner. The link prediction downside could be relevant to variety of fascinating current applications of social networks. Progressively, for example, researchers in artificial intelligence and data processing have argued that oversized organizations, such as an organization, will benet from the interactions inside the informal social network among its members. These can be serving to supplement the official hierarchy obligatory by the organization itself. Effective ways for link prediction may be wont to analyze such a social network and recommend promising interactions or collaborations that have not nevertheless been utilized inside the organization. Figure 4 can explains the process of hierarchy extraction.

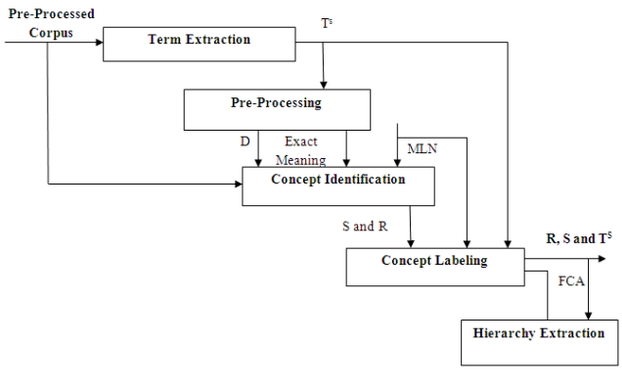

Figure 4. OBCHED process.

Formal Concept Analysis (FCA) is a method mainly used for the analysis of data in hierarchy manner. FCA can be seen as a conceptual clustering technique as it also provides intentional descriptions for the abstract concepts or data units it produces. Concept Hierarchies represent a conceptualization of a website with regard to a given corpus within the sense that they represent the relations between terms as they are utilized in the text. However corpora represent a really restricted read of the globe or a certain domain thanks to the very fact that if one thing is not mentioned, it does not mean that it is not relevant, however merely that it's not a difficulty for the text in question. The learned construct hierarchies have to be compelled regarded as approximations of the conceptualization of an exact domain.

In concept extraction process we move onto the step of hierarchy extraction using hearst pattern. Usually hearst patterns are used to discover the hypernym in the learning process. This will produce excellent result of discovering meaningful words. The main goal of our work is to automatically identify lexico-syntactic patterns indicative of hypernym. According to the taxonomic relationships we have to do the extraction process in an efficient manner.

Our first super-ordinate classifier is predicted on the intuition that unseen noun pairs square measure additional likely to be a super ordinate try if they occur within the check set with one or additional lexico-syntactic patterns found to be indicative of super ordination. We have a tendency to then produce a feature count vector for every such noun try. Our process is to discovering that dependency ways would possibly prove helpful options for our classifiers. Dependency ways are consisting of every dependency path that occurred between a minimum of five distinctive noun pairs in our corpus. To evaluate these options, we tend to make a binary classifier

for every pattern that merely classifies a noun combine as hypernym/hyponym if and providing the specific pattern happen a minimum of once for that noun combine. In our process we decide to integration of our hierarchy process into social web data. Social web data could produce the results of like Wikipedia. Wikipedia can efficiently perform the task of information retrieval. So like that our hierarchy progress should display the information about web data. The information retrieval can be integrated into the ontology learning. The social web data can produce the large amount of information of tourism places. Using these three methods such as MLN, FCA and also hearst pattern we have to produce the result in a hierarchy manner. The hierarchy order will take the order of country, organization, date, person, location, money then vehicle. The reason for choosing this order is based on the corpus. Because of here, we were chosen the corpus of lonely planet. This lonely planet can contain the details of tourism.

4. Experiments

In this division, we accomplish the wide-ranging set of experiments to scrutinize the performance of the proposed method of OBCHED framework by comparing it into the state-of-art method. The proposed OBCHED technique consists of three processes. There are Pre-Processing, Concept Identification and Concept Hierarchy Extraction. That technique can provide better results of extraction. Using this proposed technique we have to acquire the accuracy of words involved in the progress. The extraction results and some other dataset results are should be given below.

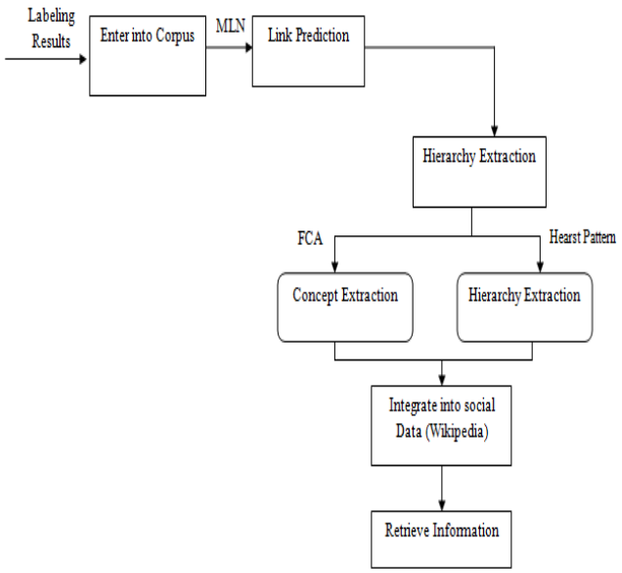

Figure 5. Concept hierarchy extraction.

4.1 Experimental Testbed

In our experimental testbed we have to use the input of lonely corpus planet dataset. This dataset are having the fields of countries, cities, cultures, organization, person and etc. This dataset is taken from the http://www.lonelyplanet.com/destinations. The reason for choosing the dataset is to examine the performance of concept extraction for more consideration. This dataset has different ways of input and details. But we are choosing only the countries and cities related details and we processed those details only limit to other properties in the dataset. Thus the dataset is suitable for our proposed OBCHED technique.

4.2 Comparison

In our experiment, we compared the proposed OBCHED technique with state-of-art method like PREHE (Probabilistic Relational Hierarchy Extraction). The previous PREHE method contains the processes of pre-processing, concept identification and hierarchy extraction. Our proposed technique also does the same processes of pre-processing, concept identification and concept hierarchy extraction with some slight changes. The changes can be highlighted first; in hierarchy extraction process the results can maintain the structural richness of web data and second integration of web data into the results of hierarchy process. So normally, pre-processing can be done with the help of some methods in learning process. The pre-processing consists of tokenization, POS tagging, Stop-Word Removal, Stemming, Lemmatization, Chunking, Syntactic Analysis and Hypernym Extraction. Here the processes of tokenization, POS tagging, Chunking and also Syntactic Analysis are all performed with the help of GATE tool. According to the

Table 1. Comparison of PREHE and OBCHED

Technique	PREHE			OBCHED		
	C_{PREHE}	C_{Hearst}	C_{FCA}	C_{OBCHED}	C_{Hearst}	C_{FCA}
Precision	6,000	10,000	10,000	12,358	12,056	12,356
Recall	130	100	105	190	140	140
F1 Measure	180	160	145	200	200	170

GATE process we get the compact results of particular document. The remaining processes are performed by our Java language. Tokenization is the process of dividing the words in the sentence. Each and every sentence had more number of tokens. The tokens are also called as keywords, phrases and some other meaningful items. The tokens are normally separated by the white spaces. The punctuation and symbols are not allowed into the tokens. According to the token generation the learning progress shall be move onto further. Next step is a parser. Here, we use Part-Of-Speech tagger used to generate the grammatical tags. This parser should remove the noun, verb and adjective present in the text. Then involves the chunking process is to be divide the words and highlight it into the result window. The set of words can be form the chunking word. After that find out the syntactic unit of words in the text. This could provide the meaningful portion of the pre-processing step and these all are done with the help of GATE tool. The GATE tool results are displayed in result window. Next, do the operation of stop-word removal at that time the letters in the words can be removed. The letters should be "is", "was", "and", "that". The separated words have taken into account the progress of taxonomic relations. Then move to the process of stemmer. Usually stemmer shall be play the operation of removing the letters of "ed", "ing", "ly" like that. So syntactic and semantic meaning of words can be identified using these steps. Next stage of pre-processing is to find out the lemmatization of each word. Lemmatization is the process of grouping together with different inflected forms of a word. Say for example the airplane is matched as the airplanes. The next process is the term weight. In this stage we are going to calculate the weight for the each term. Finally, we have to know the exact meaning of every word in the text. For that reason we can use the hypernym extraction. According to hypernym extraction the process begin to lead concept hierarchy extraction. These all are done using java language. Those pre-processing methods all are same for both state-of-art technique PREHE (Probabilistic Relational Hierarchy Extraction) and our proposed OBCHED (Ontology Based Concept Hierarchy Extraction of Web Data technique.

Next one concept is named as concept identification. In concept identification is mostly common to both the techniques. Because of in our progress we had to find the learning weight and also the inference. For learning weight decide to use the method of Markov Logic Network (MLN). To produce the inference values we have to take the alchemy process. According to that process could display an efficient value of inference. Another important concept in this technique is concept hierarchy extraction. This will differs from PREHE and OBCHED techniques. This process mainly used to relate the documents in to web. Here, first we take the hierarchy extraction process using MLN and FCA. Using MLN found the link prediction. Due to the involvement of FCA is to produce the extracted meanings in hierarchy manner. In that process we need to add the hearst pattern used to implement in word net tool to discover the hypernym of words in the corpus. After that the hierarchy extraction result can be displayed in the order of country, organization, date, person, location, money and vehicle. The order might be in efficient manner. Then finally we have to click the chunk words in the output window we can see the web related information of every possible words. This could produce the web information like Wikipedia. The information can be retrieved from the web using the process of integration of social web data. This could be related to the semantic web information.

4.3 Experimental Results

This experiment seeks to study the control of concept hierarchy extraction into the ontology learning. Figure 03. shows the Precision, Recall and F1-Measures for the proposed technique. The OBCHED technique was evaluated by comparing its output with a PREHE. For this purpose we use the data set of Lonely Planet corpus for performing the concept hierarchy extraction task. Here we evaluate three concept hierarchy extraction techniques were used in order to extract the concepts from the Lonely Planet corpus. This three extracted techniques are C_{OBCHED}, C_{HEARST}, and C_{FCA}. C_{OBCHED} is the concept hierarchy extraction using the OBCHED technique. This OBCHED technique is based on the Markov Logic Networks, which is performed by the Alchemy software packages, and also performed by the formal concept analysis.

The OBCHED technique consists of the steps of Preprocessing, Weight Learning, Inference and Concept Hierarchy Extraction. Here the pre-processing steps contains tokenization, POS tagging, chunking and syntactic analysis these all are performed by the GATE tool. Then Stemming, Stop words, Lemmatization, term weighting these can be have performed by the java code. Finally we calculate the Term weight for the pre-processing process. Then hierarchy extraction should integrate the social web data. These all process is covered by the OBCHED techniques.

The experiment results can be compared with the technique of PREHE and OBCHED. Figure 5 (a) and Figure 5 (b) can show the results of comparison of both techniques. The results can be varying with the parameter of accuracy. Our process should produce more accuracy when compared to previous technique. The accuracy can be calculated with the help of the values of precision, recall and also F1 measure. Our proposed OBCHED technique can have the best precision, recall and also F1 measure. According to those values the accuracy of every technique shall be calculated.

This paper presented a framework of OBCHED for concept hierarchy extraction which applies Markov Logic Networks for predicting the links and also provides the hierarchy results of web data integration. As a transaction between efficiency and accuracy we first proposed deterministic OBCHED technique using the pre-processing and concept hierarchy extraction. To implementing this OBCHED technique we have to use the dataset of lonely planet corpus. The dataset having the details of tourism countries, cities and temples like that. According to the dataset we take the input of words and further processes are carried out by techniques. First process of our concept hierarchy extraction is pre-processing.

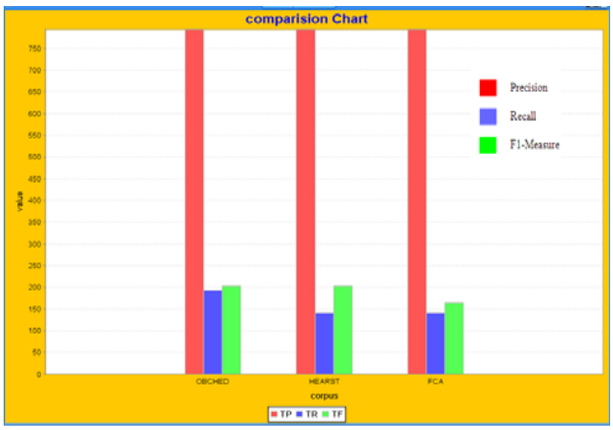

Figure 5 (a). Lonely planet corpus experimental results.

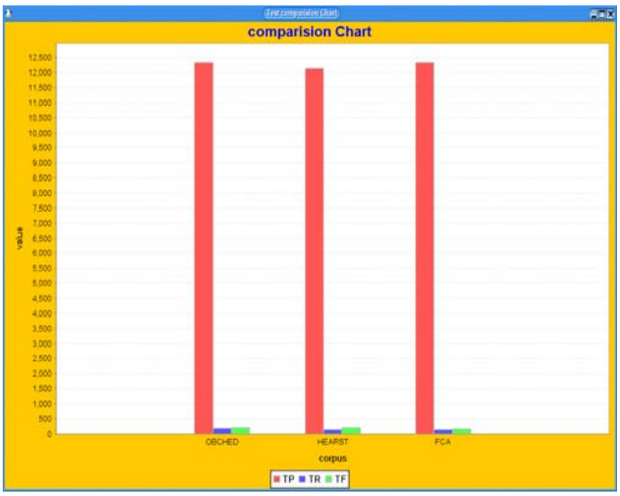

Figure 5 (b). Lonely planet corpus experimental results.

5. Discussion

Pre-processing consists of many activities. These are all placed in ontology learning progress. Pre-processing is used for extracting meaningful words from the corpus. The pre-processing activities are could be performed by tools and languages. In our process we use GATE tool for performing operations of tokenization, POS tagging, chunking and syntactic analysis. After that we had to do the activities of stop-word removal, stemming, lemmatization, term weighting and also hypernym extraction. These all are done by using Java language. Second process is concept identification. In concept identification we use MLN method to learning the weight of words. For that purpose we can use the simple weight learning method to produce the good results. The main progress include in that is to find the inference values. For finding the inference we could use alchemy process. It may be the software to produce the optimized values of every word in the corpus. Alchemy Packages also used for implement the concept identification process. Alchemy packages are used for make the perfect inference process. For implementing all this performance, we use the dataset of Lonely Planet. The final process of our OBCHED technique is the concept hierarchy extraction. In that situation, need to do the efficient way of hierarchy extraction. For this reason we predict the links in the social networks. The link prediction is process to be done with method of Markov Logic Network (MLN). Then hierarchy extraction shall be processed using formal concept analysis (FCA). In this period, we have to extract the words with meaningfully. After, that performs the hierarchy process in Hearst pattern manner. It could provide the hierarchy results in proper manner. For this purpose we should use the word net tool for extracting the hypernym words which means exact meaning. Our OBCHED technique can do the process of integrating the social data into our hierarchy manner. It could produce the efficient way of retrieving the content from the web.

6. Conclusion and Future Enhancement

In this paper we present efficient concept hierarchy extraction technique named as OBCHED. This technique could provide the best concept hierarchy extraction process with more accuracy and also the efficiency. The concept hierarchy extraction could display results of integration of web data. Thus our proposed technique gives better process. But to improve the relationship in ontology learning we can move onto the process of semantic relation. In future, the idea to make the relationship in semantic web use association rule mining for joining the relationship. To identify the non useful words we want to implement the semantic relation. Then we need to implement another process named as axiom learning. Axiom learning is an important process in learning ontology. The entire final step of the process is to implement the ontology population. Ontology population is used to analyze the population in semi automatically. This idea is decided to implement in future.

7. References

1. Hearst MA. Automatic acquisition of hyponyms from large text corpora. Proceedings of the 14th Conference on Computational Linguistics, COLING '92. 1992 Jan; 2:539–45.
2. Liben-Nowell D, Kleinberg J. The link prediction problem for social networks. Proceedings of the Twelfth International Conference on Information and knowledge Management, CIKM '03. 2003. p. 556–9.
3. Buitelaar P, Cimiano P, Magnini B. Ontology learning from text: an overview. In Applications and Evaluation. 2005. p. 3–12.
4. Koopman H, Sportiche D, Stabler E. An Introduction to Syntactic Analysis and Theory; 2013.
5. Dellschaft K, Staab S. On how to perform a gold standard based evaluation of ontology learning. Proceedings of International Semantic Web Conference, ISWC-2006; 2006.
6. Young M. The Technical Writer's Handbook. Mill Valley: CA, University Science; 1989.
7. Wong W, Liu W, Bennamoun M. Ontology learning from text: a look back and into the future. ACM Comput Surv. 2012 Aug; 44(4):36.
8. de Marneffe M-C, MacCartney B, Manning CD. Generating typed dependency parses from phrase structure parses. Department of Computing Science, Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation; 2006. p. 1–8.
9. Drumond L, Girardi R. A survey of ontology learning procedures. Proceedings Workshop on WONTO of CEUR, 427; 2008. CEUR-WS.org.
10. Karthikeyan K, Karthikeyani V. Understanding text using anaphora resolution. 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME); 2013. p. 346–50.
11. Salton G, Buckley C. Term weighting approaches in automatic text retreival. International Journal Information Processing and Management. 1998; 24:513–23.
12. Cimiano P, Hotho A, Staab S. Learning concept hierarchies from text corpora using formal concept analysis. J Artif Intell Res. 2005; 24:305–39.
13. Hofmann T. Probabilistic latent semantic analysis. Proceedings of Uncertainty in Artificial Intelligence, UAI'99; 1999. p. 289–96.
14. Chunningham H, Wilks Y, Gaizauskas RJ. A general architecture for text engineering. Proceeding at an Institute of Language, Speech and Hearing; 2012.
15. Khosravi H. Discriminative structure and parameter learning for Markov Logic Networks. Proceedings of the 25th International Conference on Machine Learning (ICML). Helsinki, Finland; 2008 Jul.
16. Drumond L, Girardi R. An experiment using Markov logic networks to extract ontology concepts from text. ACM Special Interest Group on Applied Computing; 2010. p. 1354–8.
17. Karthikeyan K, Karthikeyani V. Migrate web documents into web data. 3rd International Conference on Electronics Computer Technology (ICECT). 2011; 5:249–53.
18. Caraballo SA. Automatic Construction of a hypernym-labeled noun hierarchy from text. 99 Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics; 1999. p. 120–6.
19. Snow R, Jurafsky D, Ng AY. Learning syntactic patterns for automatic hypernym discovery. Advances in Neural Information Processing Systems (NIPS 2004). 2004 Dec 13–18.
20. Poon H, Domingos P. Sound and efficient inference with probabilistic and deterministic dependencies. AAAI'06 Proceedings of the 21st National Conference on Artificial Intelligence. 2006; 1:458–63.
21. Wu F, Weld DS. Automatically refining the wikipedia infobox ontology. 17th International Conference on World Wide Web; 2008. p. 635–44.
22. Maedche A, Staab S. Ontology learning for the semantic web. Intelligent Syst. 2011; 16:72–9.
23. Drumond L, Girardi R. Extraction only concept hierarchies from text using Markov logic. Proceedings of the 2010 ACM Symposium on Applied Computing at Federal University of Maranhao; 2010. p. 1354–8.
24. Wang T, Li Y, Bontcheva K, Wang J, Cunningham H. Automatic extraction of hierarchical relations from text.

3rd European Conference on the Semantic Web: Research and Application. 2006; p. 215–29.
25. Kavalec M, Svatek V. A study on automated relation labelling in ontology learning. In: Ontology learning from text: methods, evaluation and applications; 2005. p. 44–58.
26. Richardson AM, Domingos P. Markov logic networks. 2006; 6:1–44.
27. Dinh QT, Vrain C, Exbrayat M. Generative structure learning for Markov Logic Network based on graph of predicates. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI). 2011; 2:1249–54.
28. Satpal S, Bhadra S, Sundararajan S, Rastogi R, Sen P. Web information extraction using markov logic networks; 2011.
29. Beedkar K, Del Corro L, Gemulla R. Fully parallel inference in markov logic networks. Proceeding at Max-Planck-Institut fur Informatik. 2011; 2:373–84.
30. Satpal S, Bhadra S, Sundararajan S, Rastogi R, Sen P. Web information extraction using Markov Logic Networks. 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2011. p. 1406–14.
31. Kok S, Singla P, Richardson M, Domingos P. The alchemy system for statistical relational AI: user manual. Proceeding at University of Washington, Department of Computer Science and Engineering; 2007 Aug 3.
32. Karthikeyan K, Karthikeyani V. PROCEOL: Probabilistic relational of concept extraction in ontology learning. Internat Rev Comput and Softw. 2014; 9(4):716–26.