

MICROPOSTS' ONTOLOGY CONSTRUCTION VIA CONCEPT EXTRACTION

Beenu Yadav

Radha Govind Group of Institutions, Meerut, India

beenu_yadav@rediffmail.com

ABSTRACT

The social networking website Facebook offers to its users a feature called “status updates” (or just “status”), which allows users to create Microposts directed to all their contacts, or a subset thereof. Readers can respond to Microposts, or in addition to that also click a “Like” button to show their appreciation for a certain Micropost. Adding semantic meaning in the sense of unambiguous intended ideas to such Microposts. We can make a start towards semantic web by adding semantic annotation to web resources. Ontology are used to specify meaning of annotations. Ontology provide a vocabulary for representing and communicating knowledge about some topic and a set of semantic relationships that hold among the terms in that vocabulary. For increasing the efficiency of ontology based application there is a need to develop a mechanism that reduces the manual work in developing ontology. In this paper, we proposed Microposts' ontology construction. In this paper we present a method that extracts meaningful knowledge from microposts shared in social platforms. This process involves different steps for the analysis of such microposts (extraction of keywords, named entities and their matching to ontological concepts).

KEYWORDS

Microposts, Lexicon, Sysnset, Universal Decimal Classification (UDC), Statistically Indexed Table, Ontology, Concept Extraction, Syntatic Parsing.

1. INTRODUCTION

Social media offers a great medium for people to share their opinions and thoughts, which in turn provides a wealth of useful information to companies and their rivals, other consumers and analysts. While finding out what a single person likes and dislikes is not particularly useful on its own, the associations and conclusions that can be drawn from finding and clustering groups of people with similar interests is a veritable goldmine, going from the direct: “this group of people likes Nike products”, to the indirect: “People who like skydiving tend to be risk-takers”, to the associative: “People who buy Nike products also tend to buy Apple products”. However, the difficulty lies in accurately extracting the relevant information from the text: this is problematic even from well written sources such as online newspapers, articles and reports, but more difficult still from social media such as blogs, twitter, facebook and so on, where people use slang, do not write in full sentences or correct English, and make assumptions about the world knowledge of the reader, for example about popular culture such as books, films, news items and so on. Furthermore, it can be difficult even for a human to understand the finer concepts of the use of irony and sarcasm which is particularly present in social media, let alone for a machine.

While there are a number of sentiment analysis tools available which summarise positive, negative and neutral tweets about a given keyword or topic, these tools generally produce poor results, and operate in a fairly simplistic way, using only the presence of certain positive and negative adjectives as indicators, or simple learning techniques which do not work well on short Microposts.[4]



Figure 1. Snapshot from Facebook

An ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic entities in the domain and relations among them.[9] We develop ontology due to following reasons:

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To separate domain knowledge from the operational knowledge
- To analyze domain knowledge

1.1. Defining Ontology

Ontology is an explicit formal specification of the terms in the domain and relations among them. **Ontology** is a formal explicit description of [7]:

- **Semantic Relations** among concepts
- **Concepts** in a domain of consideration (called classes or concepts)
- **Properties** of each concept called concept description.
- **Restrictions** on properties also called facets.

A concept is an abstract, universal idea, notion or entity that serves to designate a category or class of entities, events or relations. It is a mental picture of a group of things that have common characteristics. Classes delineate concepts in the domain so they are the focus of most ontology. Semantic relations depict the collaboration of two concepts. Properties describe various features and attributes of the concept. Properties can have different restrictions such as value type, allowed values, number of values and other features of the values the property can take.

In practical terms, Ontology construction includes:

- Defining classes in the ontology,
- Relating the classes with a semantic relation,
- Arranging the classes in a taxonomic (subclass–superclass) hierarchy,
- Defining properties and describing allowed values for them,
- Filling in the values for properties for instances.

We can then create a knowledge base by defining individual instances of these classes filling in specific attribute value information and additional property restrictions.

“An ontology together with a set of individual instances of classes constitutes a knowledge base” [7].

1.2. Ontology Design

The ontology includes concepts and semantic relations with other concepts of the same domain. The concepts are described as a class, which includes their properties and restrictions on the values of the properties. The subclass inherits all the properties of the superclass but does not inherit the relationships with other classes.

1.2.1. Ontology Schema

Ontology is a specification of semantically related concept nodes. Ontology Schema can be represented by the structure of a concept node.

Concept ID: It is a unique identification of the Concept. The Concept Id is represented by any universally acceptable identification scheme. For the ease of understanding presently we are using a unique integer for concept identification such as C#110 is the Id for concept TCP/IP.

Table 1. Concept Node with Example.

Concept ID	Concept ID – C# 110
Concept Name	TCP/IP
Generic Properties	Is the most popular open-system protocol suite for communication.
Class Specific Properties	Is Robust.
Semantic Relations between Concepts	Connects: NETWORKS, Detects: ERRORS, Composed_of: LAYERS
Restrictions	Null

Concept Name: It signifies name of class corresponding to the Concept Id. Concept is a general idea formed in the mind. It is an idea about a group of things. A concept involves thinking about what it is that makes those things belong to that one group. Each word in the input text belongs to a group that identifies the concept.

Generic Properties: A set of attributes, settings and/or parameters used to define or describe an object. If a class1 has IS_A relationship with class2 it implies that it is a subclass of class2. Class1 will inherit all the properties of class2.

Class Specific Properties: Each class has its own properties defining its attributes.

Semantic Relations between concepts: This defines the relationship of a concept with others concepts. A concept may not be related with every other concept in Ontology.

Restrictions: The types of restrictions which can be imposed in an ontology can be categorized as:

- *Language Constructs:* these restrictions exist on property only and the methods to represent restrictions on property are given in Web Ontology Language and are named as Property Restrictions and Restricted Cardinality [11].

- *Restriction on Concepts:* defined by quantifiers such as double, one-fifth etc. For example, if somewhere we talk about one-third of population then ‘POPULATION’ is a concept with restriction one third. It is because we are considering only one-third population instead of entire population.
- *Restriction on Semantic Relation:* defined by conditional sentences. For example, if the sentence is, If Aditya will talk Mary, then he will meet with Alice. In this sentence, the relationship ‘will_meet’ between the concepts ADITYA and ALICE exists with the constraint ‘If Aditya will talk Mary’.

2. DEFINING VIBHAKTI PARSER

The parser verifies the grammatical correctness of the input text and identifies the ‘Vibhaktis’ or ‘Case Roles’ in the input text. So we call it “Vibhakti Parser”. The Vibhakti Parser performs two functions.

- Parsing the text
- Identifying the Vibhaktis/Case Roles

2.1. Parsing the Text

To parse the text, parser uses language grammar rules [1, 11], which are defined as production rules. This parsing examines the syntax of the text and results that text is syntactically correct or incorrect.

Parser is a collection of rules for representation of sentences in the form of production rules. The Production rules can be written as,

<simple sentence> = <subject> < verb> <complement>

The Parser has production rules for all types of sentences such as Simple sentences, Compound sentences etc.

2.2. Identifying the Vibhaktis/Case Roles

Within a sentence different nouns are connected with verb through case relationship. To identify these case relations in each language vibhaktis are used. The Paninian Grammar Framework concerns the Sanskrit language [13, 10]. However, it prescribes a generic and language independent decomposition of any sentence into eight different information carrying vibhaktis. These vibhaktis or case roles are as follows:

1. Kartaa/ Nominative - Doer of an activity or the subject.
2. Karma/Accusative - Entity that is being acted upon or the object.
3. Karan/Instrumental - Entity that is being employed to complete an act.
4. Sampradan/Dative - The chief motivation behind the action of the beneficiary subject.
5. Apadan/Ablative - Entity in Karma is separated as a consequence of the action.
6. Sambandh/Genitive - The possessor of something in the sentence.
7. Adhikaran/Locative - Place, time related to the entity at the time of action.
8. Sambodhan/Vocative - Calling upon someone – hey etc.

For example, consider the sentence,

English: The student presented the seminar of his project with projector in seminar hall.

Hindi: Student ne Apne Project ka Seminar Kaksha mein Projector se seminar ko present kiya
In this sentence,

- (i) Student – Kartaa
- (ii) Seminar – Karma
- (iii) Projector – Karan
- (iv) His Project – Sambandh
- (v) Seminar Hall – Adhikaran

2.3. Syntactic Parsing

Syntactic parsing examines the sentence syntactically and results valid sentence, if sentence is syntactically correct else results invalid sentence. The language grammar rules, which are defined in the form of production rules, are used to parse the text [1, 5]. For representation of sentences, production rules are described in the parser. It includes representation for all types of sentences. Input sentences are parsed by defined sentence structure rules and when it sets to any one of the rules then that sentence is proved to be syntactically correct.

Example:

S1: I called him but he gave me no answer.

- <Simple Sentence> <Conjunction> <Simple Sentence>
- <I> <called him> <Conjunction> <he> <gave me no answer>
- <subject1> <predicate1> <Conjunction> <subject2> <predicate2>
- <subject1> = <nominative personal pronoun>
- <predicate1> = <V> <complement>
- = <Vpast> <object>
- <subject2> = <nominative personal pronoun>
- <predicate2> = <V> <complement>
- = <Vpast> <indirect object> <object>

2.4. Vibhakti Parsing

The Vibhakti Parser parses the syntactically correct sentence to identify the vibhaktis, states, verbs and others elements. The rule base is made for determination of each of them. After remodeling we apply the following rules and identify Vibhaktis, States, etc.

2.4.1. Rule Base

For identification of Vibhaktis/Case roles

1. Subject of the sentence is identified as Kartaa Vibhakti.
2. If the subject has pronoun then Parser replace it with the corresponding noun, it is identified as Kartaa Vibhakti.
3. Rest of the Vibhkatis are identified from complement of the sentence.

- a. If complement has an object(direct/indirect) then it is Karam Vibhakti.
 - b. In case of pronoun object before determining Vibhakti, Parser substitutes it with its respective noun.
4. The vibhaktis are identified by preposition in the prepositional phrase.
5. In prepositional phrase if
- a. Preposition is “ Main verb+ to + NP ” → Karam Vibhakti
 - b. Preposition is “by, with, from” → Karan Vibhakti
 - c. Preposition is “for, to + Vinf” → Sampradaan Vibhakti
 - d. Preposition is “from*, by*” → Apadaan Vibhakti
 - e. Preposition is “of, to*” → Sambandh Vibhakti
 - f. Preposition is “at, in, on, above” → Adhikaran Vibhakti

from* => ‘from’ when used with some special verbs that indicate separations such as fell, break or some phrases as fell down etc. then it is categorized as Apadaan Vibhakti else it is Karan Vibhakti.

by* => ‘by’ when used with some special verbs that indicate separations such as fell or some phrases as letting off etc. then it is categorized as Apadaan Vibhakti else it is Karan Vibhakti.

to* => ‘to’ when used in the form other than as explained in ‘a’ and ‘c’ then it is Sambandh Vibhakti.

We have categorized some prepositions for identifying Vibhaktis/Case roles. In a similar manner this categorization of prepositions can be enhanced by working on more prepositions such as compound prepositions, phrase prepositions.

For identification of Verbs

1. Verbs or verb phrases in the sentence represent actions.

For identification of States

1. Some sentences represent state rather than actions; the state is identified as property of the subject.

For identification of Other Elements

1. The conditional sentences impose restrictions on either the verbs or the property. The ‘if’ clause or ‘when’ clause of such sentences is added to all the relations.
2. The quantifiers are added as restrictions to the noun/noun phrase that will be further identified as concepts in the construction of ontology.

2.5. Formation of Vibhakti Table

The Vibhakti Parser generates the Vibhakti Table of the input document on applying vibhakti parsing rules on syntactically correct simple sentences. Vibhakti Table has columns for Verb of the sentence, one for property of Kartaa in the sentence, seven for Vibhaktis/case roles of sentence. Using the above defined rules, Vibhakti Parser frames a Vibhakti Table for given text/document.

2.5.1. Steps for Framing Vibhakti Table

1. Each sentence is processed for syntactic correctness by using Production rules defined above in Syntactic Parsing section.
 - a. If the parsed sentence (after remodeling, if any) is valid in grammatical sense then it undergoes Vibhakti Parsing.
 - b. Else Syntactic Parsing is interrupted and the subsequent sentence is treated as the next input for parsing.
2. Each syntactically valid simple sentence is scanned for identifying noun phrases, verbs or prepositional phrases. As the Parser encounters any one of these then using Vibhakti Parsing rules, Vibhaktis/case roles, verbs and properties are determined.
3. The determined vibhaktis, verbs and properties are simultaneously fed into the respective cell of Vibhakti Table.

The pictorial representation of Vibhakti Parser can be delineated in figure 2.

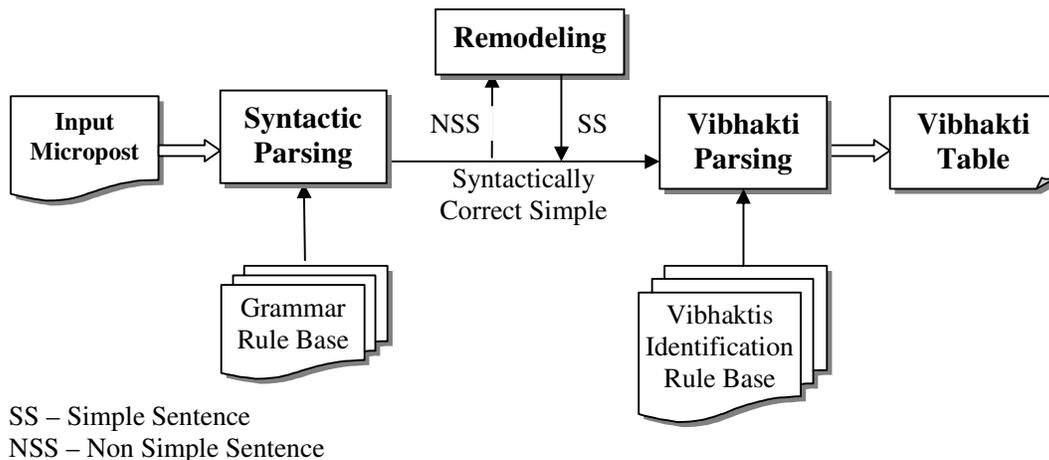


Figure 2. Vibhakti Parser

Example:

➤ *The lecture was focused on the problem of unemployment.*

Table 2. Vibhakti/Case Role Table

S. No.	Verb	Karta	Karman	Karan	Sampradan	Apadan	Sambandh	Adhikaran	Property
1	Was focused	The lecture					of unemployment	on the problem	

3. CONCEPT EXTRACTOR

The concept extractor is a module designed for the determination of concepts of the ontology. The nouns and the noun phrases are the keys which form concepts in the ontology [8, 2, 12]. For this purpose we scale some existing linguistic resources according to our requirement and design new components using some existing resources.

3.1. Lexicon

A Lexicon is a repository of words and knowledge about those words. A lexicon is a list of words together with additional word-specific information. It is a list of corresponding terminology in different languages, usually locale, industry or project specific [3].

Lexicon used for microposts ontology builder, incorporates-

1. Collection of Words
2. Unique Id(s) respective to each word: It is a Universal Decimal Classification (UDC) that uniquely identifies the concepts. The UDC(s) are determined from the SynSet table.
3. The category to which the word belongs based on classification of concepts is attached. The classification of concepts is given in the forthcoming section.

The word extracted from text/document for the identification of concept may or may not be matched with any word from the collection of words in Lexicon. When word does not match with any entry of Lexicon directly then morphology [6] is used.

For Example, words like Networks, Leaves etc., are not found in Lexicon. In these words morphemes are –

1. Network, -s
2. Leaf, -ves

To identify UDC(s) for these words, these words are analyzed as sequence of morphemes so that one of the word forms gets matched in Lexicon.

3.2. SynSet Table

The SynSet Table is a table developed for the identifications of words possessing the same meaning. It is the collection of synonymous words with the attribute set. The unique identification number is given to the set of words that have the identical meaning and such set identify the unique concept.

To each unique concept we give UDC (Universal Decimal Classification) identification as its unique identification number. The UDC is the world's foremost multilingual classification scheme for all fields of knowledge. An advantage of this system is that it is infinitely extensible, and when new concepts are introduced, they need not disturb the allocation of numbers to the existing concepts [13].

In every language there are some words that express multiple meanings when used in different contexts. The exact meaning of such word is determined from the context of sentence in which the word is used. For this purpose we attach an attribute set with such words in the SynSet Table. In case when a word with different meaning in different contexts is encountered then the attribute set is exploited for the identification of exact word.

Each row in the SynSet table consists of three columns.

- a) The first column of every row has UDC.
- b) The second column has synonymous words having the same concept.
- c) The third column has Attribute Set. The motivation for this is to provide a framework for finding semantically sensible concept of a multi-contextual word provided by the Lexicon.

For Example,

Table 3. SynSet Table.

UDC	Synonym Set	Attribute Set
5/6:523.31.12	Space, Area, Volume, Region	one, two, or three dimensional; bounded, occupied by objects
5/6:528.93	Space, Outer Atmosphere	Related to solar system, beyond the earth's atmosphere, boundless

3.3. Statistically Indexed Concept Table

Extracting concepts requires a technique that can retrieve the appropriate concepts from documents of any subject domain. Statistical indexing technology is accurate enough to compute extraction of concepts [2].

The Vibhakti Parser extracts the units, such as noun phrases; they can be used to depict concepts by computing their frequency across the document. The indexing can be accomplished by computing the statistical frequency of extracted noun phrases within each document in a collection. The Statistically Indexed Concept Table is constructed by entering each noun phrase with its UDC. The UDC is determined from Lexicon and SynSet table. The noun/noun phrases, their UDC identification and their count altogether shape the Statistically Indexed Concept Table. Example:

Table 4. The Statistically Indexed Concept table.

Row No.	Nouns/Noun Phrases	Frequency	UDC
1	TCP/IP, TCP and IP	7	681.324.003
2	Local Area Network, LAN, LAN operations	3	681.324.001
3	Computer Networks	5	681.324

The frequency index of each noun/noun phrase changes while the document is read. The frequency index of the table corresponding to each concept determines the validated concepts of the ontology.

3.4. Concept Extraction Method

The functioning of Concept Extractor is shown pictorially in figure 3.

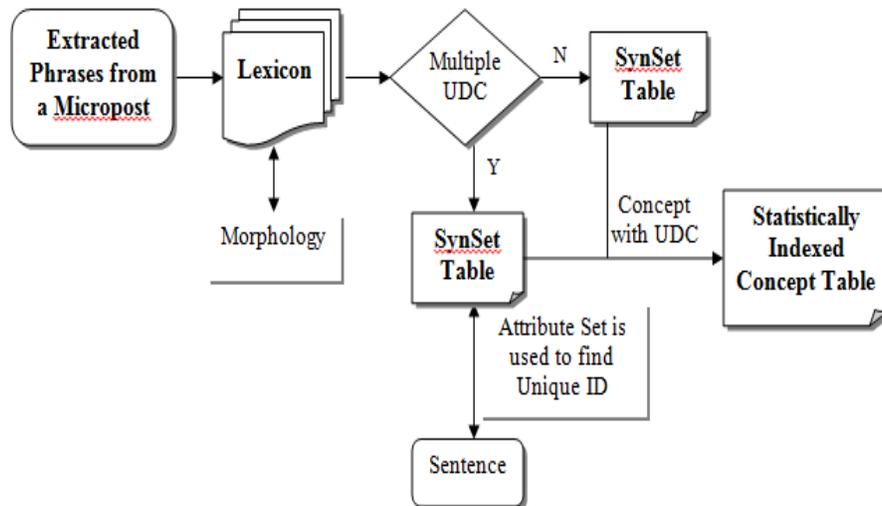


Figure 3. Functioning of Microposts' Concept Extractor

This section outlines the methodology for figuring out the concepts for an ontology using above illustrated components and resources. Lexicon and SynSet Table are used to develop the Statistically Indexed Concept table, which is used to determine the concepts for the ontology. The step wise procedure is given as:

1. The word/phrase is extracted from the sentence to determine its concept.
2. This extracted word/phrase is mapped to the Lexicon. The Lexicon consists of UDC(s) relative to each word. These Unique Id(s) is used to find the concept(s) from SynSet table.
3. There may be more than one Unique Id corresponding to each word, which indicates that the word is used in different senses or contexts. The context of the extracted word is resolved using Attribute Set which is defined in SynSet Table.
4. The Unique Id found by the concept extractor is searched into the Statistically Indexed Concept Table. If it is found then the frequency corresponding to that Unique Id is increased by one and the extracted noun/noun phrase is appended to the Noun/Noun Phrase column.
5. For each extracted word/phrase
 - a) If the extracted word/phrase has one UDC in the Lexicon then this identification is fed into Statistically Indexed Concept Table.

- b) Otherwise the complete sentence is read and the SynSet table is referred to determine its unique concept. With the help of Attribute Set and the sentence, the unique concept of the word/phrase is determined. Corresponding to the unique concept the UDC is identified and fed into the Statistically Indexed Concept Table.
- c) Unique Id and the extracted noun/noun phrase are made as a new entry into the table with the frequency 1.

4. MICROPOSTS' ONTOLOGY BUILDER

The Microposts' ontology builder is an endeavor to reduce the manual effort in the construction of ontology. This saves the time and thus efficiency of the work will be increased. We have explained the Vibhakti Parser which is a pillar of the Microposts' auto ontology builder. The second pillar of Microposts' auto Ontology Builder is Concept Extractor. Vibhakti Parser with the Concept Extractor is integrated to develop ontology of any document. The forthcoming sections explain methodology for Microposts' ontology construction.

4.1 Architecture of Microposts' Ontology Builder

The development of Microposts' Ontology Builder is an approach to the automatic construction of ontology from the existing information resources.

The input document is passed to the Vibhakti Parser for the syntactic checking of the sentences and the noun/noun phrases identified during parsing are fetched by Concept Extractor to construct Statistically Indexed Concept Table. The Vibhakti table is constructed using the rule base of Vibhakti Parser. The concepts for the ontology under construction are determined from the Statistically Indexed Concept Table. These concepts and the Vibhakti Table, concurrently gives the structure to the ontology.

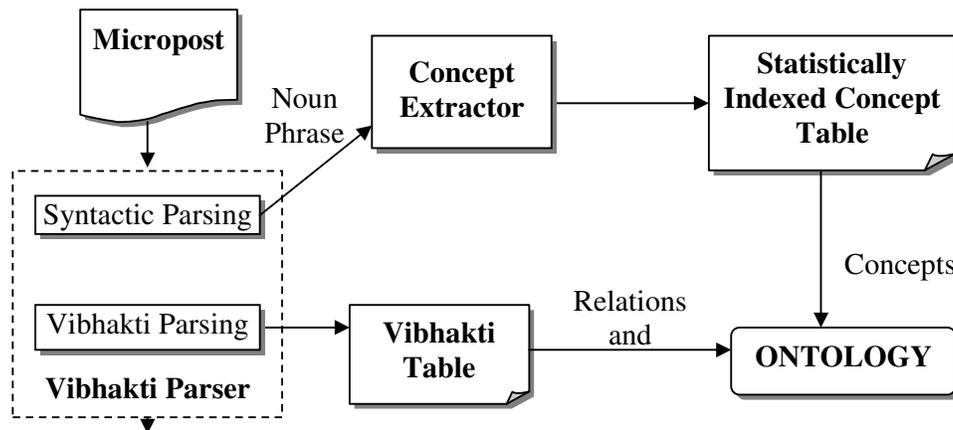


Figure 4. Architecture of Microposts' Ontology Builder

4.2 Functioning of Microposts' Ontology Builder

4.2.1 Algorithm

Step 1: Parsing and Remodeling of the Sentence

The input text/document is parsed for checking the grammatical correctness of the sentences and simultaneously the non simple sentences encountered are converted into simple sentences. The result of syntactic parsing and remodeling is syntactic tagged sentence and it is directly used for vibhakti parsing and for concept identification.

Step 2: Vibhakti Parsing and Concept Identification

The syntactically parsed sentence is used by Vibhakti Parser and Concept Extractor. On every tagged part of the sentence,

- the rules of vibhakti parsing are applied to identify the vibhaktis and
- simultaneously the noun/noun phrase are passed to concept extractor for the identification of concepts.

Step 3: Construction of Statistically Indexed Concept Table

The noun/noun phrase of the parsed sentence is used to identify concepts. The concept extractor uses Lexicon and SynSet Table to generate Statistically Indexed Concept Table, which contain the Unique Id and Frequency of occurrence corresponding to each concept.

Step 4: Construction of Vibhakti Table

The noun/noun phrase in the corresponding vibhakti column forms a concept and has an unique record in Statistically Indexed Concept Table. The noun/noun phrase and their respective Row No. retrieved from the Statistically Indexed Concept Table are fed into the vibhakti table.

The verbs of the sentence define the action, which is inserted into verb column of the Vibhakti Table.

The states are represented by properties, which is inserted into property column of the table. The conditional sentences from the text impose the constraint on the action so it is written into the verb column of the row.

The quantifiers, multipliers etc. impose the restrictions on the nouns, which are fed into the Vibhakti column corresponding to that concept.

The Vibhakti Table identifies the vibhaktis, verbs, restrictions and properties such as dates, digits, units, formulae etc. Hence, Concept Extractor determines concepts and Vibhakti Parser parses each sentence of the text to construct the Vibhakti Table, which is ideally developed for the microposts' construction of ontology.

Step 5: Approving the Concepts

Since there are many concepts in the text of which ontology is to be made, out of all those some selected concepts will form the ontology, such selected concepts are approve concepts. Concepts are approved based on following procedure.

To approve concepts we refer to the statistically indexed concept table. This table has concepts with their UDC and the frequency of occurrence of concept in the input document. The concepts with the frequency index greater than the threshold value are approved concepts of the ontology to be built. The threshold value is determined beforehand. This value is application dependent and based on the criterion specified by the user.

Step 6: Microposts' Ontology Formation

Ontology is a specification of semantically related concept nodes. Ontology Schema can be represented by the structure of a concept node. For each approved concept identified from Kartaa Vibhakti we write a concept structure. A concept node structure includes:

- Concept ID
- Concept Name
- Properties
- Semantic Relations
- Restrictions

The Kartaa column of each row of the Vibhakti table is scanned subsequently to check that the noun/noun phrase is an approved concept. The elements that give structure to concept node relative to the approved concept are identified from the row of Vibhakti table. Otherwise the row of the Vibhakti table under consideration is not scanned further and the next row is scanned.

Concept ID and Concept Name

The concept Id is unique UDC identification taken from Statistically Indexed Concept Table. The name of the concept structure is the concept name, which is the highly significant noun/noun phrase retrieved from the respective column of the Statistically Indexed Concept Table.

Properties and Semantic Relations

The properties are written in sentential form. The properties that have a subset-superset type structure such as Is_a, Kind_of, Type_of followed by noun only or an adjective and a noun only then it forms a subset relationship which is included in semantic relations of the concept node. The semantic relations in the ontology are identified from the vibhakti table with the help of verbs and the prepositions. For the determination of relationship here we state the semantics for writing the relations between concepts.

- a) The relationship is determined from the main verb and the preposition.
- b) If the 'Sampradan' column of the row under consideration has verb then the relationship is identified by the verb in this column instead of combination of main verb and the preposition.
- c) If the row has an entry in 'Karam' column along with entries in other columns except 'Sampradan' then the relationship is identified by the combination of main verb, entry in 'Karam' column and the preposition.
- d) Relation between concepts that form Self loop is ignored unless the concepts have the restrictions/facets attached to them.

There may be instances when the approved concept is related to rejected concept but relationship between such concepts is included in the concept structure of the ontology built automatically.

Restrictions

1. Restriction on Semantic Relationship: The restriction on semantic relation is written with relationship in the concept structure.
2. Restriction on Concept: Constraints on concepts are portrayed in two forms.
 - Based on the approved concept which has its concept structure.
 - If all the relations and properties are with same restricted concept then we write restriction with the concept name.
 - Else we categorize the relations and properties based on the restriction on the concept. The restriction is written with the categories.
 - Based on the unapproved concept to which the concept node is related with a semantic relation.
 - The restriction is written with the unapproved concept.

Similarly, the entire table is scanned and the ontology of the text is constructed.

5. CONCLUSIONS

This paper proposed a technique to extract concepts from plain text to build ontologies. The extraction is based on existing linguistic resources like lexicon and synset. A Universal Decimal Classification is associated with each concept to classify the concepts. The Syntactic Parsing is to be done using Vibhakti Parser to preprocess the text and convert the compound and complex sentences into simpler sentences. The noun/noun phrases are extracted from the preprocessed text which are input to the concept extractor which extracts the potential nouns as the concepts. It uses Statistically indexed table is generated with the validation of the concept in text. Those concepts are extracted which are occurring most frequently in the text. This technique helps to extract the concepts from the Microposts’.

REFERENCES

- [1] Basic English Sentence Structures, <http://www.scientificpsychic.com/grammar/enggram3.html>.
- [2] Bruce R. Schatz , IEEE Computer (2002), “The Interspace: Concept Navigation Across Distributed Communities”, <http://www.canis.uiuc.edu/archive/papers/interspace.computer.pdf>.
- [3] Lexicon, <http://en.wikipedia.org/wiki/Lexicon>.
- [4] Michael Hartl, “Ruby on Rails Tutorial”, <http://ruby.railstutorial.org/chapters/user-microposts>
- [5] Modern English Grammar, <http://papyr.com/hypertextbooks/grammar/>.
- [6] Morphology (Linguistics), http://en.wikipedia.org/wiki/Morphology_%28linguistics%29
- [7] Natalya F. Noy and Deborah L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology”, <http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf>.

- [8] Nuala A. Bennett, Qin He, Conrad Chang, Bruce R. Schatz, “Concept Extraction in the Interspace Prototype”, <http://www.canis.uiuc.edu/archive/techreports/UIUCDCS-R-99-2095.pdf>, Technical Report, Digital Library Initiative Project, University of Illinois at Urbana-Champaign, 1999.
- [9] Ontology Working Group, <http://mged.sourceforge.net/ontologies/index.php>.
- [10] Sanskrit Grammar: Noun Cases, http://www.everything2.com/index.pl?node_id=1017898.
- [11] OWL Web Ontology Language Reference, W3C Recommendation 2004, <http://www.w3.org/TR/2004/REC-owl-ref-20040210>.
- [12] Spela Vintar, Paul Buitelaar Martin Volk, “Semantic Relations in Concept-Based Cross-Language Medical Information Retrieval”, <http://www.dcs.shef.ac.uk/~fabio/ATEM03/vintar-ecml03-atem.pdf>, 2003.
- [13] UDC Consortium, <http://www.udcc.org/>.

Author

Beenu Yadav, has done B.C.A., M. Sc. (Computer Science) and currently, pursuing M.Tech (Computer Science & Engg.) from MTU, Noida. Presently, working as Assistant Professor at College of Professional Education, Meerut, India. And also, a certified Java Professional – SCJP & SCWCD. Published three papers in various International Journals, one is published in Springer, one published in proceedings of a National Conference and one is communicated. Presented two papers, one in National Conference & one in International Conference.

