Domain Specific Ontology Extractor For Indian Languages

Brijesh Bhatt *Pushpak Bhattacharyya*
Center For Indian Language Technology, Indian Institute of Technology, Bombay
brijesh@cse.iitb.ac.in, pb@cse.iitb.ac.in

ABSTRACT
We present a k-partite graph learning algorithm for ontology extraction from unstructured text. The algorithm divides the initial set of terms into different partitions based on information content of the terms and then constructs ontology by detecting subsumption relation between terms in different partitions. This approach not only reduces the amount of computation required for ontology construction but also provides an additional level of term filtering. The experiments are conducted for Hindi and English and the performance is evaluated by comparing resulting ontology with manually constructed ontology for Health domain. We observe that our approach significantly improves the precision. The proposed approach does not require sophisticated NLP tools such as NER and parser and can be easily adopted for any language.

KEYWORDS: Ontology extraction, k-partite graph, wordnet, concept hierarchy.

Proceedings of the 10th Workshop on Asian Language Resources, pages 75–84,
COLING 2012, Mumbai, December 2012.

1 Introduction

Ontology is defined as 'Explicit specification of conceptualization' (Gruber, 1993). As a knowledge representation formalism, ontologies have found a wide range of applications in the areas like knowledge management, information retrieval and information extraction.

As manual construction of ontology is a cumbersome task, many supervised and unsupervised techniques have been proposed to automatically construct ontology from the unstructured text. The ontology learning process involves two basic tasks- domain specific concept identification and constrution of concept hierarchy. Most of the existing algorithms extract relevant terms from the documents using various term extraction methods (Ahmad et al., 1999; Kozakov et al., 2004; Sclano and Velardi, 2007; Frantzi et al., 1998; Gacitua et al., 2011) and then construct ontology by identifying subsumption relations between terms.

Identifying top level concepts and creating a good concept hierarchy are the major challenges involved in the ontology learning tasks. As noted by Fountain and Lapata (2012), 'Most of the existing approaches construct flat structure rather than a taxonomy. Also, the automatically constructed ontologies often create false association between terms and result in erroneous concept hierarchy (Zhou, 2007).

In order to handle the above mentioned issues, we propose a graph-based ontology learning algorithm. Our approach is based on the information content of the term. 'Terms with high information content remain lower in the concept hierarchy and terms with low information content remain higher in the concept hierarchy' (Resnik, 1999). Caraballo and Charniak (1999) have shown that the term frequency is a good indicator of determining specificity of a term.

We divide the initial set of terms into different partitions based on the term frequency and then construct k-partite graph by finding subsumption relation between the terms of different partitions. This approach not just reduces the amount of computation required for ontology construction but also provides an additional level of term filtering. This early identification of hierarchy creates a better taxonomic structure and avoids false association between the terms.

The proposed approach combines evidences from linguistic patterns and WordNet (Fellbaum, 1998) to detect subsumption relation. The patterns used in the system are generic and can be used across languages. Wordnets of Indian languages are linked with each other and English WordNet through a common index (Bhattacharyya, 2010), which makes it possible to share concept definitions across languages.

Following are the major features of the proposed system:

- Ontology extraction process is completely unsupervised and does not require any human intervention.
- The lexical patterns used in the algorithm are generic and can work for any language.
- Proposed graph partition based algorithm not only requires less computation than the existing clustering techniques but also reduces false association between terms.
- The proposed system does not require sophisticated NLP techniques such as NER or parser and can be used for resource constrained languages.

The paper is organized as follows: section 2 describes related work, proposed algorithm is described in section 3 and section 4 discusses experiment and evaluation.

2 Related work

As noted by Leenheer and Moor (2005), ‘No matter how expressive ontologies might be, they are all in fact lexical representations of concepts’. The linguistic basis of formal ontology is such that a significant portion of domain ontology can be extracted automatically from the domain related texts using language processing techniques. The problem of ontology learning is well studied for English. However, to the best of our knowledge no such efforts have been made so far for Indian languages.

Ontology learning approaches can be divided into three categories: heuristic based, statistical and hybrid techniques. Heuristic approach (Hearst, 1992; Berland and Charniak, 1999; Girju et al., 2003) primarily relies on the fact that ontological relations are typically expressed in language via a set of linguistic patterns. Hearst (1992) outlined a variety of lexico-syntactic patterns that can be used to find out ontological relations from a text. She described a syntagmatic technique for identifying hyponymy relations in free text by using frequently occurring patterns like ‘*NP0 such as NP1, NP2, . . . ,NPn*’. Berland and Charniak (1999) used a pattern-based approach to find out part-whole relationships (such as between car and door, or car and engine) in a text. Heuristic approaches rely on language-specific rules which cannot be transferred from one language to another.

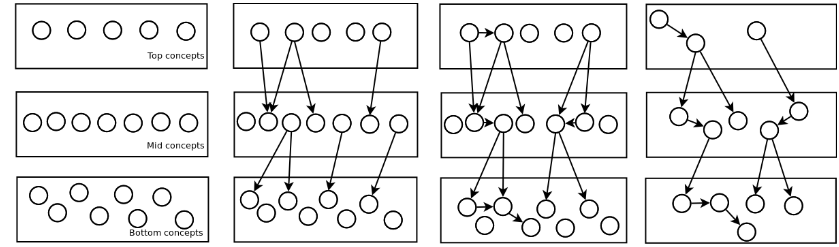
Statistical approaches model ontology learning as a classification or clustering problem. Statistical methods relate concepts based on distributional hypothesis (Harris, 1968), that is ‘similar terms appear in the similar context.’ Hindle (1990) performed semantic clustering to find semantically similar nouns. They calculated the co-occurrence weight for each verb-subject and verb-object pair. Verb-wise similarity of two nouns is calculated as the minimum shared weight and the similarity of two nouns is the sum of all verb-wise similarities. Pereira et al. (1993) proposed a divisive clustering method to induce noun hierarchy from an encyclopedia.

Hybrid approaches leverage the strengths of both statistical and heuristic based approaches and often use evidences from existing knowledge bases such as wordnet, wikipedia, etc. Caraballo (1999) combined the lexico-syntactic patterns and distributional similarity based methods to construct ontology. Similarity between two nouns is calculated by computing the cosine between their respective vectors and used for hierarchical bottom-up clustering. Hearst-patterns are used to detect hypernymy relation between similar nouns. In a similar approach, Cimiano et al. (2005) clustered nouns based on distributional similarity and used Hearst-patterns, WordNet (Fellbaum, 1998) and patterns on the web as a hypernymy oracle for constructing a hierarchy. Unlike (Caraballo, 1999), the hypernymy sources are directly integrated into the clustering, deciding for each pair of nouns how they should be arranged into the hierarchy. Domínguez García et al. (2012) used wikipedia to extract ontology for different languages.

Like Cimiano et al. (2005), we follow a hybrid approach and construct a concept hierarchy using distributional similarity, patterns and WordNet. However, instead of performing top-down or bottom-up clustering, we pose ontology learning as a k-partite graph construction problem. We use term frequency to determine the position of a concept in the hierarchy. Ryu and Choi (2006) also used term frequency as a measure of domain specificity, but instead of partitioning they combined term frequency and distributional similarity to construct hierarchy. Other method similar to our work is proposed in Fountain and Lapata (2012). Fountain and Lapata (2012) proposed a graph based approach that does not require a separate term extraction step. However, their approach works with a predefined set of seed terms. Our approach is completely unsupervised and does not require any human intervention or predefined seed terms. Term frequency based partition provides early detection of the top level concepts and provides an additional level of term filtering.

3 Algorithm

The proposed algorithm poses ontology learning as a k-partite graph learning problem. The ontology graph is defined as a directed acyclic graph G(V, E), where V is a set of concept nodes and E is a set of relation edges. The proposed algorithm initially divides terms into different partitions and then constructs ontology by relating terms across partitions. The process involves three tasks, i.e., preprocessing, k-partite graph creation and concept hierarchy generation. Figure 1 represents the overall taxonomy learning process.



A. Create Initial Partition (High frequency terms in Top concepts Low frequency terms in Bottom concepts)

B. Link terms across partition based on similarity. Identify 'IS-A' relation using patterns and wordnet

C. Expand Hierarchy within partition

D. Refine hierarchy by restructuring edges. Remove unconnected nodes

Figure 1: Taxonomy Learning Process

3.1 Preprocessing

This module extracts domain specific terms from the text corpus. The corpus is processed by performing morph analysis, POS tagging and stop word removal. Then lexical pattern $(NP) * (NP)$ is applied to extract key phrases from the corpus. Relevance of the key term in the corpus is calculated by counting the frequency of the term. Terms are filtered out using weirdness measure (Ahmad et al., 1999). Feature vector for each term is created by including co-occurring nouns, verbs and adjectives.

3.2 Initial Partition Creation

For each input term, concept node is created by calculating term frequency, feature vector and wordnet synsets. A concept node v ∈ V is defined as <t, tf, sid, v >. where, t = lexeme for the concept, tf = Frequency of the term in corpus, sid = wordnet sense for the concept, v = feature vector for the concept. Once the concept nodes are created, the node set V is divided into three subsets based on frequency of the concept. High frequency terms are placed in top partition and low frequency terms are placed in bottom partition.

3.3 K-partite Graph Construction

This module constructs bipartite graph by finding relation edges between nodes of different partitions. The process involves two steps: calculate semantic relatedness between concepts and identify the type of relation i.e. subsumption.

As shown in algorithm 1, semantic relatedness between each concept pair (V_i, V_j), where $V_i \in C_i$ and $V_j \in C_j$, is measured by calculating cosine similarity between the feature vectors. The feature vector for a concept is constructed by including co-occurring nouns, verbs and adjectives. The weight of a feature is calculated using pointwise-mutual-information. A relation edge E(V_i , V_j) is created if the similarity value is found greater than the predefined threshold value.

Algorithm 1 *Link Partitions*

```
$C_1$ := concept nodes in Partition 1; m:= $|C_1|$
$C_2$ := concept nodes in Partition 2; n:= $|C_2|$
e := edge set; $|e|$:=0
for $each C_{1i} \in C_1 and C_{2j} \in C_2$ do
    similarity := $\frac{\vec{C_{1i}} \cap \vec{C_{2j}}}{|C_{1i}| * |C_{2j}|}$
    if $similarity > Threshold$ then
        create edge $e_l$ := ($C_{1i}$, $C_{2j}$)
    end if
end for
```

Evidences from wordnet and lexico-syntactic patterns are used to detect name of the relation between semantically related concepts. Different relations identified during this phase are, subsumption (*e.g. pneumonia-disease*), neighbor (*e.g. malaria-pneumonia*) and similar (*e.g. procedure-process*).

Two patterns are used to detect *subsumption* and *neighbor* relations. Head word heuristic (Cimiano, 2006) based pattern $(NP) * (NP)$ is used to identify subsumption relation. As per head word heuristic $(NP1)(NP2)$ implies $(NP2)$ subsumes $(NP1NP2)$, *e.g. health-care program is-a-kind-of program*. This pattern often creates many false positives. False positives are reduced by applying frequency constraint; if $(NP1)(NP2)$ is in the low frequency partition and there exist term $(NP2)$ in high frequency partition then $(NP2)$ is parent of $(NP1)(NP2)$.

Various Hearst patterns to detect subsumption relation are, *'such NP as (NP,)∗ (and|or) NP'*, *'NP such as (NP,)∗ (and|or) NP'*, *'NP (, NP)∗ (,) or other NP'*, etc. Existing ontology extractors use many such patterns to detect subsumption relation. However, these patterns are specific to a language and in order to use the system for multiple languages we need to code these patterns for all languages. Instead, we generalize this to a single pattern, $((NP) * (NP)(and|or|,)) * (NP)(NP)$. As per this pattern, if two or more noun phrases appear in the sentence separated by commas or conjunctions then these noun phrases are neighbors/co-hyponyms. For example, in a sentence '*such diseases as malaria and pneumonia....*' the original Hearst patterns can detect two subsumption relation edges (*malaria IS-A disease and pneumonia IS-A disease*), while our generalized pattern detects one neighbor/co-hyponymy relation (*malaria 'is neighbor' pneumonia*).

In addition to patterns, wordnet is also used to detect subsumption and synonymy relation between the terms. For the given pair of terms, synonymy is identified if they occur in the same synset for at least one sense pair. If the two terms are not synonyms, subsumption is investigated between the terms. If one term is the hypernymy of another, sense pair for which the hypernymy distance is smallest is returned as subsumption edge.

Algorithm 2 *Refine Hierarchy*

```
Graph(V, E)
while No change in edges do
  V := Concept Set; k:= $|V|$; E:= Edge Set; m:= $|E|$;
  for $eachE_i \in E$ do
    $V_1$ = source concept of $E_i$; $V_2$ = target concept of $E_i$
    if $E_i$ is $synonym$ then
      merge concept $V_1$ and $V_2$
    end if
    if $E_i$ is $neighbor$ then
      Create edges from parent of $V_1$ to $V_2$ and vice versa
    end if
  end for
  for $eachV_i \in V$ do
    $V_p$ := parent of $V_j$; p := $|V_p|$;
    for $eachV_{p_q}, V_{p_r} \in V_p, V_{p_q} \neq V_{p_r}$ do
      if $V_{p_q}$ is parent of $V_{p_r}$ then
        remove edge between $V_{p_q}$ and $V_j$
      end if
    end for
    $V_c$ := children of $V_j$; p := $|V_c|$
    for $eachV_{cq}, V_{cr} \in V_c, V_{cq} \neq V_{cr}$ do
      if $V_{cq}$ is parent of $V_{cr}$ then
        remove edge between $V_j$ and $V_{p_r}$
      end if
    end for
  end for
end while
```

3.4 Concept Hierarchy Creation

This process refines the k-partite graph constructed in the previous step and creates a concept hierarchy. A random walk through the nodes of the graph is performed to refine the relation hierarchy. Two major tasks performed during this phase are, (1) ‘neighbor’ and ‘synonymy’ edges constructed during previous phase are removed and new ‘subsumption’ edges are constructed accordingly. (2) The resulting subsumption graph is refined to improve hierarchy. Algorithm 2 describes the process.

During this process, the nodes linked with synonymy edge are merged and neighbor edges are removed. For each neighbor edge (V_i, V_j) subsumption edges are created from V_k to V_j, if V_k is parent of V_i and from V_l to V_i, if V_l is parent of V_j. The subsumption hierarchy is refined by investigating subsumption relation between each pair of concept for which there is a common subsuming node.

Finally, all concept nodes that do not have any incoming or outgoing edges are removed. ‘k-partiteness’ of the graph is ensured by checking that each weakly connected subgraph contains nodes from atleast two partitions. A weakly connected subgraph $G'(V', E')$ is removed if it does not contain at least one edge $e'(v_1, v_2) \in E'$ for which v_1 and v_2 are in different partition. This provides an additional level of term filtering and the relation edges which are not representative of the domain are removed.

4 Experiments and Observations

In order to evaluate performance of the system, we conducted our experiments on health corpus for two languages, Hindi and English. The details of the corpus is shown in table 1. We constructed ontology in both languages using our partitioned algorithm and without partition (similar to agglomerative clustering). We investigated relation across the layers to check which evidences are useful at which layer and compared the resulting ontology with a hand crafted ontology.

Corpus	No. of Sentences	No. of Terms
English Health	15589	16498
Hindi Health	16002	14794

Table 1: corpus details

4.1 Layer wise evidence detection

Table 2 shows the source of evidence across layers. As shown in the table, in top partition the relation between concepts is detected more often using wordnet while in bottom partition evidences from lexico-syntactic patterns are more frequent. This is consistent with our hypothesis that top level concepts are general concepts and can be found in wordnets.

Partition	English Health		Hindi Health	
	LSP	WORDNET	LSP	WORDNET
Top	75	214	0	85
Mid	238	1012	23	313
Bottom	342	313	297	91
Mid-Bottom	420	1094	138	310
Top-Mid	137	1050	5	399
Top-Bottom	131	549	39	191

Table 2: Layer wise evidence

4.2 Comparison with gold standard

The quality of the ontology constructed is evaluated by comparing it with the hand crafted ontology. The lexical precision and recall is calculated using following formula,

Recall = $|c1 \bigcap c2|/c2$

Precision = $|c1 \bigcap c2|/c1$

where c1 is the set of concept in automatically constructed ontology and

c2 is the set of concepts in hand crafted gold standard.

Table 3 shows the precision and recall for both cases: with partition and without partition. As shown in table 3, the precision is higher for partitioned algorithm.

	Precision	Recall	F-Score
English-Health No Partition	0.69	0.83	0.75
English-Health Partition	0.75	0.7298	0.7298
Hindi-Health No Partition	0.81	0.604	0.6789
Hindi-Health Partition	0.9251	0.7679	0.8387

Table 3: Evaluation against hand crafted ontology

Conclusion

We have presented a novel graph based algorithm for domain specific ontology extraction. Our approach is unsupervised and does not require any human intervention. The proposed system can be easily adopted for any language. Using our algorithm, we constructed 'health domain ontology' from English and Hindi text corpora and the resulting ontology is compared against a manually constructed ontology. It is observed that partitioning improves the precision without sacrificing F-Score. We also observe that the high frequency terms remain at the top level in the ontology and definition for these terms are often found in wordnet, while lexico-syntactic patterns are found more often in low frequency terms. Our future aim is to include automatic extraction of non-taxonomic relations between concepts.

References

Ahmad, K., Gillam, L., Tostevin, L., and Group, A. (1999). Weirdness indexing for logical document extrapolation and retrieval (wilder). In *The Eighth Text REtrieval Conference*.

Berland, M. and Charniak, E. (1999). Finding parts in very large corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 57–64, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bhattacharyya, P. (2010). Indowordnet. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 120–126.

Caraballo, S. A. and Charniak, E. (1999). Determining the specificity of nouns from text. In *Proceedings SIGDAT-99*, pages 63–70.

Cimiano, P. (2006). *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Cimiano, P., Pivk, A., Schmidt-Thieme, L., and Staab, S. (2005). Learning taxonomic relations from heterogeneous sources of evidence. *Ontology Learning from Text: Methods, Evaluation and Applications*.

Domínguez García, R., Schmidt, S., Rensing, C., and Steinmetz, R. (2012). Automatic taxonomy extraction in different languages using wikipedia and minimal language-specific information. In *Proceedings of the 13th international conference on Computational Linguistics and Intelligent Text Processing - Part I*, CICLing'12, pages 42–53, Berlin, Heidelberg. Springer-Verlag.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

Fountain, T. and Lapata, M. (2012). Taxonomy induction using hierarchical random graphs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 466–476, Montréal, Canada. Association for Computational Linguistics.

Frantzi, K. T., Ananiadou, S., and Tsujii, J.-i. (1998). The c-value/nc-value method of automatic recognition for multi-word terms. In *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '98, pages 585–604, London, UK, UK. Springer-Verlag.

Gacitua, R., Sawyer, P., and Gervasi, V. (2011). Relevance-based abstraction identification: technique and evaluation. *Requir. Eng.*, 16(3):251–265.

Girju, R., Badulescu, A., and Moldovan, D. (2003). Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of HLT/NAACL-03*, pages 80–87.

Gruber, T. R. (1993). Towards principles for the design of ontologies used for knowledge sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands. Kluwer Academic Publishers.

Harris, Z. (1968). Mathematical structures of language. John Wiley Sons.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 539–545.

Hindle, D. (1990). Noun classification from predicate-argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, ACL '90, pages 268–275, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kozakov, L., Park, Y., Fin, T.-H., Drissi, Y., Doganata, Y. N., and Cofino, T. (2004). Glossary extraction and utilization in the information search and delivery system for ibm technical support. *IBM Systems Journal*, 43(3):546–563.

Leenheer, P. D. and Moor, A. D. (2005). Context-driven disambiguation in ontology elicitation. In *Context and Ontologies: Theory, Practice, and Applications. Proc. of the 1st Context and Ontologies Workshop, AAAI/IAAI 2005*, pages 17–24. AAAI Press.

Pereira, F., Tishby, N., and Lee, L. (1993). Distributional clustering of english words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res. (JAIR)*, 11:95–130.

Ryu, P.-M. and Choi (2006). Taxonomy learning using term specificity and similarity. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 41–48, Sydney, Australia. Association for Computational Linguistics.

Sclano, F. and Velardi, P. (2007). Termextractor: a web application to learn the shared terminology of emergent web communities.

Zhou, L. (2007). Ontology learning: state of the art and open issues. *Information Technology and Management*, 8:241–252. 10.1007/s10799-007-0019-5.