Expectation Maximization Algorithm for Domain Specific Ontology Extraction

Brijesh Bhatt and Pushpak Bhattacharyya

Center for Indian Language Technology
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India
brij.s.bhatt@gmail.com pb@cse.iitb.ac.in

Abstract. Learning ontology from unstructured text is a challenging task. Over the years, a lot of research has been done to predict ontological relation between a pair of concepts. However all these measures predict relation with a varying degree of accuracy. There has also been work on learning ontology by combining evidences from heterogeneous sources, but most of these algorithms are ad hoc in nature. In this paper we investigate wide range of evidences to predict relation between a pair of concepts and propose a standardized Expectation Maximization algorithm to construct domain specific ontology. The proposed approach is completely unsupervised and does not require any seed terms or human intervention. In addition, the proposed approach can also be easily adopted for any language. We have conducted our experiments for two languages Hindi and English and for two domains Health and Tourism. The average F-Score observed in all experiments is above 0.60.

1 Introduction

Ontology is defined as 'Explicit specification of conceptualization' [13]. As a knowledge representation formalism, ontologies have found a wide range of applications in the areas like knowledge management, information retrieval and information extraction. Domain and application specific ontologies play a crucial role in semantic web applications.

As manual construction of ontology is a cumbersome task, a lot of research is being done to automatically construct ontology from the unstructured text. In general, ontology learning process involves two basic tasks- domain specific concept identification and construction of concept hierarchy. Most of the existing algorithms extract relevant terms from the documents using various term extraction methods [19, 23, 10, 11] and then construct ontology by identifying subsumption relations between terms.

Detection of subsumption relation is a core task of ontology extraction. Over the years, a number of approaches have been proposed to detect subsumption between a pair of concepts. These approaches can be divided into three categories: Pattern based, Knowledge based, and statistical. Pattern based approaches rely

pp. 11–21; rec. 2015-01-25; acc. 2015-02-25

on seminal work of [15] who outlined a variety of lexico-syntactic patterns that can be used to find out 'IS-A' from a text. Statistical approaches are based on distributional hypothesis that is 'similar terms appear in the similar context' [14]. Availability of general purpose lexical resources like, WordNet [9], Wikipedia etc. gave rise to knowledge based approach. Many researchers have used *Hypernymy* relation of WordNet and category pages of Wikipedia to detect subsumption relation between a pair of concepts [8].

While all these approaches detect subsumption relation with a reasonable precision, they are quite different and have their own strengths and weaknesses. Pattern based approach relies on language specific patterns and hence does not cater to different languages. Patterns are also not very frequent so this approach may not work well for a small corpus. Statistical approaches primarily detects 'semantic relatedness' between a pair of concept but detection of relation still remains a challenging task. Besides, the result of these approaches are often contradictory.

There has been efforts to learn domain ontology by combining different approaches but these methods are mostly ad hoc. In addition, most of the existing work in ontology learning is done for English language and often uses sophisticated Natural Language Processing (NLP) tools and resources. In absence of such tools and resources, it becomes difficult to adopt these methods for resources constrained languages.

The key challenges in the task of ontology learning are as follows,

- Development of a standardized approach to combine different subsumption detection technique for better ontology extraction
- Development of an approach which can be easily adopted for different languages; particularly resource constrained languages

In order to address the above mentioned challenges, we propose an ontology extraction framework which learns ontology from heterogeneous sources and does not rely on sophisticated NLP tools and resources. In this paper, we first analyze various techniques for subsumption detection and then propose an Expectation-Maximization algorithm to learn ontology. The key contributions of our work are as follows,

- The proposed system is completely unsupervised and does not require any labeled data or human intervention
- The proposed approach does not rely on any language specific technique or resource and hence can be adopted for any language

The remaining of the paper is organized as follows, section 2 presents a survey of existing work, section 3 describe the algorithm to learn ontology, section 4 describes experiments and observations and section 5 provides.

2 Related Work

Ontology learning approaches can be divided into three categories: heuristic based, statistical and hybrid techniques. Heuristic approach [15, 2, 12] primarily relies on the fact that ontological relations are typically expressed in language via a set of linguistic patterns. [15] 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*'. [2] 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 [14], that is 'similar terms appear in the similar context.' [17] 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. [21] 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. [3] 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, [5] clustered nouns based on distributional similarity and used Hearst-patterns, WordNet [9] and patterns on the web as a hypernymy oracle for constructing a hierarchy. Unlike [3], the hypernymy sources are directly integrated into the clustering, deciding for each pair of nouns how they should be arranged into the hierarchy. [8] used Wikipedia to extract ontology for different languages.

Most of the hybrid approaches which combine pattern based approach and statistical approach, are ad-hoc in nature. They first use the statistical clustering to group the terms and then uses knowledge base (e.g. WordNet) and patterns to identify relation. In this work, we are proposing a starndardized Expectation Maximization algorithm that merges evidences from different sources. We treat each measure as a feature to detect relation ship. Apart from these, the proposed algorithm detects three relations, synonymy, hypernymy (subsumption) and neighbor (co-hyponymy).

3 Algorithm

Most of the existing algorithms to learn ontology from heterogeneous sources of evidences are ad hoc in nature and use languages specific measures (e.g. lexical patterns for English, English WordNet etc) to detect relation between a pair of concepts. We also follow hybrid approach where we combine statistical, pattern based and knowledge based methods to learn ontology; but unlike other existing systems we choose evidences which can be adopted for any languages and we use a standardized Expectation-Maximization based algorithm which can be adopted for different sets of evidences. More specifically, we use evidences from different sources as features and use Expectation-Maximization algorithm to learn relation between a pair of concepts. The process of ontology learning is as described in Fig 1

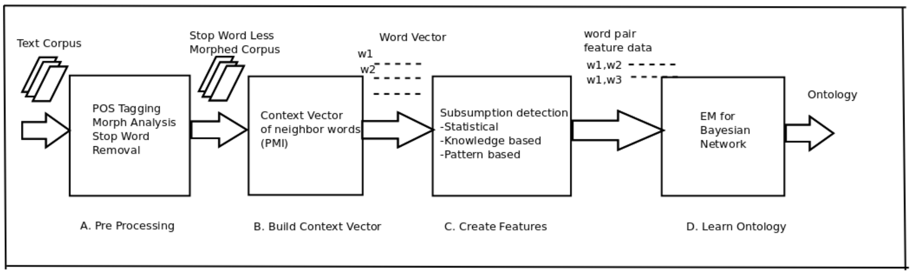

Fig. 1: Ontology Learning Process

3.1 Pre-Processing

The input text is processed using POS tagger and morph analyzer. The stop words and junk words are removed. The output of pre-processing step is pos tagged, stop word less corpus. For English we have used *morpha* [20] morph analyzer and Stanford POS tagger [1], for Hindi we have used morph analyzer and pos tagger developed at IITB [2].

3.2 Context Vector Construction

Key terms from the corpus are extracted using pattern based method. Lexical pattern $(NP)^*(NP)$ is applied to extract key phrases from the corpus. Relevance of the key term is calculated by counting the frequency of the term. Terms are filtered out using weirdness measure [1].

In order to compare concepts, we construct the context for a word using bag of word approach. Feature vector for each term is created by including co-occurring nouns, verbs and adjectives. Co-occurrence is calculated using Point-wise Mutual Information [4] measure.

3.3 Feature Construction

Features are the indicator of semantic relation between a pair of words. In order to construct concept hierarchy we detect subsumption (hypernymy), synonymy and co-hyponymy relations. Various measures we used to detect these relations are as shown in Table 1.

[1] http://nlp.stanford.edu/software/tagger.shtml
[2] http://www.cfilt.iitb.ac.in/Tools.html

Table 1: Features for ontology Learning

	feature	Description	formula
f_1	Cosine Similarity	Cosine similarity between word w_1 and w_2 is calculated by comparing the vectors of words.	$cosine(w_1, w_2) = \frac{\vec{V}(w_1) \cdot \vec{V}(w_2)}{\lvert\vec{V}(w_1)\rvert\lvert\vec{V}(w_2)\rvert}$
f_2	Weeds Precision	This measure quantifies the weighted inclusion of the features of a term w_1 within the features of a term w_2. [24, 18].	$WeedsPrec(w_1, w_2) = \frac{\sum_{f \in F(w_1) \bigcap F(w_2)} w_1(f)}{\sum_{f \in F(w_1)} w_1(f)}$
f_3	cosWeeds	This measure corresponds to the geometrical average of Weeds Precision and cosine similarity between words w_1 and w_2	$cosWeeds = \sqrt{cosine(w_1, w_2). WeedsPrec(w_1, w_2)}$
f_4	ClarkeDE	This measure is a close variation of Weeds Precision, proposed by [6].	$ClarkeDE(w_1, w_2) = \frac{\sum_{f \in F(w_1) \bigcap F(w_2)} min(w_1(f), w_2(f))}{\sum_{f \in F(w_1)} w_1(f)}$
f_5	Frequency Ratio	We use frequency ratio to measure degree of generality of a word. The measure is based on following hypothesis, 'A more general term appears more frequently in the corpus, while a more specific term appears less frequently' [22]	$fratio(w_1, w_2) = \frac{f(w_1)}{f(w_2)}$
f_6	Head Word heuristic Pattern	This pattern finds hypernymy relation from noun phrase. e.g. "Heritage Hotel" is a "Hotel"	(NP)*NP is hyponym of (NP)
f_7	Neighbor Pattern	This pattern detects neighbor (Co-hyponymy) relation. e.g. Delhi, Mumbai, Calcutta are cities.	$((NP)*(NP)(CC\|,))*(NP)$
f_8	WordNet hypernym	This formula calculates probability of hypernymy by consulting WordNet	$\frac{hypernym(w_1, w_2)}{totalRelation(w_1, w2)}$
f_9	WordNet Synonym	This formula calculates probability of synonymy by consulting WordNet	$\frac{synonym(w_1, w_2)}{totalRelation(w_1, w2)}$
f_{10}	WordNet Neighbor	This formula calculates probability of co-hyponymy by consulting WordNet	$\frac{co-hyponym(w_1, w_2)}{totalRelation(w_1, w2)}$

3.4 Ontology Learning

Output of feature construction step is as shown in Table 2. As shown in Table 2, the output of different features does not match and they often contradict. Our aim is to predict the correct relation between the pair of words using the observed values of features. Each of this feature gives a hint about possible relation between a pair of words. We assume that the relation y is the common cause that triggers one or more features. We model a Bayesian network as shown in the figure 2. The random variable Y corresponds to the relation between the pair of words and $X_1...X_d$ correspond to the feature vector components $f_1, f_2, ..., f_d$. $X_1...X_d$ are observed variables and Y is the hidden variable e.g. the relation that we want to predict.

Table 2: Example DataSet

i	**Word Pair**	**f$_1$**	**f$_2$**	**f$_3$**	**f$_4$**	**f$_5$**	**f$_6$**	**f$_7$**	**f$_8$**	**f$_9$**	**f$_{10}$**	**f$_{11}$**	**f$_{12}$**	**y$_1$**	**y$_2$**	**y$_3$**
1	haemorrhagic fever - dengue haemorrhagic fever	1	1	1	1	1	1	0	0	0	0	0	0	?	?	?
2	leptospirosis-kalaazar	2	0.39	0.24	0.24	0.30	0	0	1	0	0	0	0	?	?	?
3	transplant - transplantation	1.91	0.20	0.14	0.14	0.17	0	0	0	0.33	0	0	0	?	?	?
.	...	...	...	...	...	...	...	...	...	...	...	...	....	?	?	?
n	cannabis - marijuana	1.6	0.18	0.14	0.14	0.16	0	0	0	0.25	0	0.25	0	?	?	?

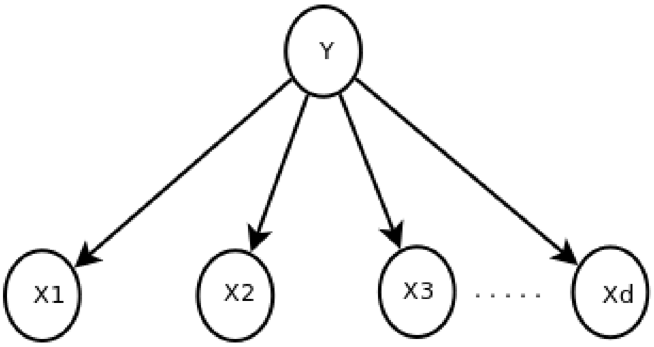

Fig. 2: Bayesian Model for Predicting Relation

The problem of predicting correct relation between a pair of word can now be posed as a Bayesian network learning problem. Given the observed variables X (features) our aim is to predict the hidden variable Y (relation e.g. synonymy, hypernymy, co-hyponymy).

Joint probability of the network can be expressed as shown in equation 1.

$$P(Y = y, X_1 = x_1, X_2 = x_2, ...X_d = x_d) = P(Y = y) \prod_{j=1}^{d} P(X_j = x_j | Y = y) \tag{1}$$

Here, $P(Y = y)$ and $P(X_j = x_j|Y = y)$ for $j = 1, 2..., d$ are network parameters. Let's define parameter vector θ as a vector consisting values of these parameters.

Likelihood Since the value of hidden variable Y (relation between word pair) is not known, probability of an example can be calculated by marginalizing over all possible values of label Y, as shown in the following equation 2.

$$P(x) = \sum_{y=1}^{k} P(Y = y) \prod_{j=1}^{d} P(X_j = x|Y = y) \tag{2}$$

For the complete training set containing n examples, likelihood can be calculated using the equation 3.

$$L(\theta) = \sum_{i=1}^{n} log \sum_{y=1}^{k} (P(Y^{(i)} = y) \prod_{j=1}^{d} P(X_j^{(i)} = x|Y^{(i)} = y)) \tag{3}$$

Our goal is to predict the network parameters θ that maximize likelihood of the data.

EM Algorithm Since the relation label is not known, we use Expectation-Maximization algorithm to iteratively estimate the value of parameters θ that maximizes the likelihood. For the training set consisting of vectors of observed variable $x^{(i)}$ and hidden variables $y^{(i)}$ for $i = 1...n$ and the parameter vector θ, the EM algorithm starts by randomly choosing the initial parameter values θ_0. At each iteration value of hidden variable $Y^{(i)}$ is calculated as a function of the training set and the previous parameter values θ_{t-1}; and the new parameter values θ_t are updated using the observed variables and previously estimated hidden variables. [16, 7].

E-Step For the given value of θ, E-Step calculates probability of hidden variable for each example X_i using equation 5.

$$\delta(y|i) = p(Y^{(i)} = y|X^{(i)};\theta_{t-1}) \tag{4}$$

$$p(Y^{(i)} = y|x^{(i)};\theta_{t-1}) = \frac{P(Y^{(i)} = y)\prod_{j=1}^{d} P(X_j^{(i)} = x_j/Y^{(i)} = y)}{\sum_{y=1}^{k}\left(P(Y^{(i)} = y)\prod_{j=1}^{d} P(X_j^{(i)} = x_j/Y^{(i)} = y)\right)} \tag{5}$$

Expectation of hidden variable Y is then calculated using equation 6

$$E(Y = y) = \sum_{i=1}^{n} \delta(y|i) \tag{6}$$

M-Step Based on the value of hidden variable estimated in E-Step, M-Step calculates new parameter values as shown in equation 7 and 9.

$$P(Y = y)^t = \frac{\sum_{i=1}^{n} \delta(y|i)}{n} \tag{7}$$

$$where, n = total number of examples \tag{8}$$

$$p(X_j = x|Y = y)^t = \frac{\sum_{i=1:X_j^i=x}^{n} \delta(y|i)}{\sum_{i=1}^{n} \delta(y|i)} \tag{9}$$

4 Experiments and Observations

We have carried out our experiments for two domains, health and tourism and for two languages, English and Hindi. We choose English to compare results against benchmark and We choose Hindi as a resource constrained language. We have used untagged text corpus for both domains Health and Tourism [3]. Table 3 shows the details of the corpora.

We preprocessed the English corpus using Stanford POS tagger and Morpha morph analyzer and Hindi corpus with CFILT POS Tagger. After extracting key terms and building context vector the features are extracted as described earlier.

In order to measure performance of Individual features we measure precision of top 100 word pair for each measures. Table 4 summarizes the results. As shown in the table pattern based and Knowledge based measures perform much better than the statistical measures. However, the relation tuple detected by all these measures are often different. WordNet based measures detect subsumption between more frequent and general terms while pattern based and statistical measures detect relation between low frequency terms.

[3] The corpora are available at $http://www.cfilt.iitb.ac.in/wsd/annotated\_corpus/$

Table 3: corpus details

Domain	Language	Total Sentences	Words
Health	Hindi	25000	61000
	English	25000	69000
Tourism	Hindi	48000	89000
	English	59000	121000

Table 4: precision@100 for Individual features

Corpus	Weeds Precision	cosWeeds	clarkeDE	Pattern	WordNet
English Health	0.36	0.39	0.32	.67	.72
English Tourism	0.42	0.48	0.48	.69	.70
Hindi Health	0.34	0.40	0.38	.70	.68
Hindi Tourism	0.40	0.38	0.36	0.65	0.68

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 = $|y_1^r \bigcap y_2^r|/y_2^r$ and Precision = $|y_1^r \bigcap y_2^r|/y_1^r$

where y_1^r is the set of relation r detected in automatically constructed ontology and y_2^r is the set of relation r detected in hand crafted gold standard. We have run experiments to detect four relations i.e. *hypernymy, synonymy, neighbors (co-hyponymy) and no relation.* The precision (P), recall (R) and F-Score (F) for each domain and for each language are summarized in the table 5

Table 5: Experiment Results

Domain	Language	Hypernymy			Neighbor			Synonym			No Relation			Average		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Health	Hindi	0.59	0.84	0.69	0.66	0.65	0.66	0.42	0.81	0.55	0.82	0.46	0.59	0.62	0.69	0.66
	English	0.62	0.85	0.72	0.61	0.70	0.65	0.58	0.75	0.66	0.81	0.50	0.62	0.66	0.70	0.68
Tourism	Hindi	0.54	0.85	0.66	0.62	0.60	0.61	0.3	0.63	0.41	0.8	0.43	0.55	0.57	0.63	0.6
	English	0.56	0.79	0.65	0.54	0.65	0.59	0.63	0.75	0.68	0.76	0.47	0.58	0.62	0.66	0.64

As shown in table 5, the average F-Score observed for both languages and both domains is between 0.60 to 0.70. The performance is reasonably good considering the fact that the algorithm is completely unsupervised and does not rely on language specific evidences. Synonymy detection does not perform very well for Hindi language as resources for Hindi are not as rich as that of English.

5 Conclusion

In this paper we have presented an unsupervised algorithm for domain ontology extraction. The proposed approach does not rely on language specific resources or tools and can be easily adopted for any language. The proposed algorithm consults evidences from different sources e.g. statistical measures, knowledge based measures and pattern based measures and predicts relation between a pair of words. While most of the existing ontology learning algorithms focuses only on hypernymy / IS-A relation detection, our algorithm also detects synonymy and co-hyponymy, thus it provides a more refined ontology by merging words that represent similar concepts. The proposed Expectation-Maximization framework is general enough to accommodate other measures as features or to detect more relations (e.g. whole-part etc). We have conducted experiments for two languages and two domains and average precision and recall was higher than 0.60.

References

1. Ahmad, K., Gillam, L., Tostevin, L., Group, A.: Weirdness indexing for logical document extrapolation and retrieval (wilder). In: The Eighth Text REtrieval Conference (1999)
2. Berland, M., Charniak, E.: Finding parts in very large corpora. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. pp. 57–64. ACL '99, Association for Computational Linguistics, Stroudsburg, PA, USA (1999)
3. Caraballo, S.A.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics. pp. 120–126 (1999)
4. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Comput. Linguist. 16(1), 22–29 (mar 1990)
5. Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S.: Learning taxonomic relations from heterogeneous sources of evidence. Ontology Learning from Text: Methods, Evaluation and Applications (2005)
6. Clarke, D.: Context-theoretic semantics for natural language: An overview. In: Proceedings of the Workshop on Geometrical Models of Natural Language Semantics. pp. 112–119. GEMS '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)
7. Collins, M.: The naive bayes model, maximum-likelihood estimation, and the em algorithm (2013), lecture notes
8. Domínguez García, R., Schmidt, S., Rensing, C., Steinmetz, R.: 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. pp. 42–53. CICLing'12, Springer-Verlag, Berlin, Heidelberg (2012)
9. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)

10. Frantzi, K.T., Ananiadou, S., Tsujii, J.i.: 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. pp. 585–604. ECDL '98, Springer-Verlag, London, UK, UK (1998)
11. Gacitua, R., Sawyer, P., Gervasi, V.: Relevance-based abstraction identification: technique and evaluation. Requir. Eng. 16(3), 251–265 (sep 2011)
12. Girju, R., Badulescu, A., Moldovan, D.: Learning semantic constraints for the automatic discovery of part-whole relations. In: Proceedings of HLT/NAACL-03. pp. 80–87 (2003)
13. Gruber, T.R.: Towards principles for the design of ontologies used for knowledge sharing. In: Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers, Deventer, The Netherlands (1993)
14. Harris, Z.: Mathematical structures of language. John Wiley Sons (1968)
15. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th International Conference on Computational Linguistics. pp. 539–545 (1992)
16. Heckerman, D.: Learning in graphical models. chap. A Tutorial on Learning with Bayesian Networks, pp. 301–354. MIT Press, Cambridge, MA, USA (1999)
17. Hindle, D.: Noun classification from predicate-argument structures. In: Proceedings of the 28th annual meeting on Association for Computational Linguistics. pp. 268–275. ACL '90, Association for Computational Linguistics, Stroudsburg, PA, USA (1990)
18. Kotlerman, L., Dagan, I., Szpektor, I., Zhitomirsky-geffet, M.: Directional distributional similarity for lexical inference. Nat. Lang. Eng. 16(4), 359–389 (oct 2010)
19. Kozakov, L., Park, Y., Fin, T.H., Drissi, Y., Doganata, Y.N., Cofino, T.: Glossary extraction and utilization in the information search and delivery system for ibm technical support. IBM Systems Journal 43(3), 546–563 (2004)
20. Minnen, G., Carroll, J., Pearce, D.: Applied morphological processing of english. Nat. Lang. Eng. 7(3), 207–223 (sep 2001)
21. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of english words. In: Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics. pp. 183–190 (1993)
22. Resnik, P.: 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 (1999)
23. Sclano, F., Velardi, P.: Termextractor: a web application to learn the shared terminology of emergent web communities (2007)
24. Weeds, J., Weir, D.: A general framework for distributional similarity. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing. pp. 81–88. EMNLP '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003)