Efficient Extraction of Ontologies from Domain Specific Text Corpora

Tianyu Li
University of British Columbia
Vancouver, BC, Canada
lty419@cs.ubc.ca

Pirooz Chubak
University of British Columbia
Vancouver, BC, Canada
pchubak@cs.ubc.ca

Laks V.S. Lakshmanan
University of British Columbia
Vancouver, BC, Canada
laks@cs.ubc.ca

Rachel Pottinger
University of British Columbia
Vancouver, BC, Canada
rap@cs.ubc.ca

ABSTRACT

Extracting ontological relationships (e.g., ISA and HASA) from free-text repositories (e.g., engineering documents and instruction manuals) can improve users' queries, as well as benefit applications built for these domains.

Current methods to extract ontologies from text usually miss many meaningful relationships because they either concentrate on single-word terms and short phrases or neglect syntactic relationships between concepts in sentences.

We propose a novel pattern-based algorithm to find ontological relationships between complex concepts by exploiting parsing information to extract multi-word concepts and nested concepts. Our procedure is iterative: we tailor the constrained sequential pattern mining framework to discover new patterns. Our experiments on three real data sets show that our algorithm consistently and significantly outperforms previous representative ontology extraction algorithms.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Ontologies, Ontology extraction

1. INTRODUCTION

An ontology is a specification of conceptualizations in a specific domain [2]. An ontology typically includes, at a minimum, concepts and hierarchical relationships among them. Two of the fundamental hierarchical relationships are ISA, which asserts a class-subclass or a class-instance relationship, and HASA, which asserts a whole-part relationship.

Building an ontology from unstructured text can bridge the gap between human-readable data and machine-readable knowledge. However, general-purpose ontologies (e.g., WordNet) have limited coverage, particularly in specialized fields, where jargon and terminology can have different meanings from their senses in a more general domain. For example, in the architecture domain, "foreign materials" usually refers to external substances on the surface of a piping system; Quite different from the meaning of foreign in "foreign nationals" which may be found in a legal document.

Extracting rich and complex concepts along with ontological relationships among them as accurately as possible is the main goal of this work. We develop an iterative pattern-based algorithm called LASER (for LArge ScalE Relation extraction system). The central paradigm used by LASER is an iterative framework of starting with seed patterns that signal an occurrence of (ISA or HASA) relations, which are used to extract instances of relations in the corpus. They are in turn used to induce more patterns from the corpus and so forth. At every stage, the extracted patterns and instances are scored, reflecting their degree of reliability. In summary, we make the following contributions:

- A novel algorithm to extract complex concepts having ISA/HASA relationships from parsed text and to deal with nested noun phrases.
- Starting with a list of seed patterns from the ontology extraction literature [8, 1, 7].
- Adaptation of a sequential pattern mining framework to find frequent patterns that signal ISA/HASA relationships.
- A detailed set of experiments over three real datasets.

The rest of this paper is structured as follows: Section 2 describes the related work. We discuss the LASER approach and the algorithms in Section 3 and discuss our experiments and lessons learned in Section 4. Finally, Section 5 concludes and discusses future work.

2. RELATED WORK

This paper focuses on automatically building terminological ontologies from domain-specific text corpora. Many approaches build ontologies consisting of single words [3, 15] or short common compounds [17, 4]. Indeed, few works allow for longer and complex terms to be concepts, generally because they have a very low frequency of occurrence compared to short ones. Drymonas et al. [6] allow more complicated noun phrases using statistical measures and agglomerative

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

clustering. Navigli and Velardi [12, 13] proposed Word-Class Lattices (WCLs) which can be used to iteratively extract a set of class-subclass relationships, resulting in a Hypernym Graph (WCL-HG) from which a domain taxonomy is induced. Compared to our approach, WCLs require manual tagging of definition sentences and use POS tagging, which does not perform well on identifying boundaries of complex and nested phrases.

There exist clustering-based ontology extraction algorithms that produce prototype-based ontologies (E.g. Caraballo [3]). Pantel and Ravichandran [15] first cluster words from a text corpus; each cluster forms a concept. They extract concept names by searching for syntactic patterns such as "*concept* apposition-of *instance*" and "*concept* such as *instance*". Finally, they assign labels to each cluster according metrics such as to the co-occurrence frequencies of concepts and instances. . They create ISA relationships between the concept and all instances in the cluster. However, this kind of ISA relationship is necessarily confined to one level hierarchy, while our work produces a complex multi-level concept hierarchy.

Pure pattern-based methods such as Pantel and Pennacchiotti [14], often iteratively interleave pattern discovery and instantiation until the reliability drops below a threshold. These methods suffer from low recall. ESPRESSO [14] is a system that given a small set of seed instances for a particular relation, learns lexical patterns, applies them to extract new instances, and then uses the web to filter and expand the instances. We adapt their scoring mechanism in the iterative process, but instead of starting with seed instances, we bootstrap from seed patterns. Also, we extract instances from patterns substantially differently from ESPRESSO as well: we rely on deep parsing information to get richer concepts that cannot be identified by regular expression matching over data obtained from shallow parsing. Our pattern finding is also generalized to be less restrictive and more expressive than ESPRESSO. Additionally, we provide a novel formulation of pattern discovery as a constrained sequential pattern mining problem.

Guided Hierarchical Clustering (GHC) [4] is a hybrid approach that first calculates the similarity between a set of given input terms based on syntactic dependency features in the corpus. Then, using an agglomerative clustering algorithm, it picks the most similar pair of terms in the remaining list of pairs to be clustered, and uses WordNet, Hearst patterns in the corpus, and the world wide web to position them in the growing ontology. Unlike GHC, our approach finds relationships before concepts. We start from patterns that indicate relations, and then get concepts from there, thus not requiring terms as input.

3. ONTOLOGY EXTRACTION

LASER uses an iterative process. It takes as input the preprocessed corpus consisting of a set of text documents, with each word tokenized. Additionally, it takes in a set of *seed patterns*, i.e., lexico-syntactic templates such as Hearst patterns [8] that imply ISA/HASA relationships. We define a *Subsumption Candidate Instance Pair* (SCIP) as a pair of noun phrases x, y such that they are involved in a class-subclass or class-instance (ISA) or a whole-part (HASA) relationship, and denote it SCIP(x, y). In this section, we provide a description of modules in the architecture of LASER. More detailed description of these modules can be found in [10].

3.1 ISA/HASA Pattern Instantiation

This module takes a list of known patterns suggesting ISA or HASA relationships and applies the patterns to the input corpus to find sentences matching the patterns. Previous works [4, 14] usually find pattern instances by matching each POS tagged sentence with regular expressions. Such a strategy has the following limitations: (1) simple POS tag rules may identify the wrong noun phrase because the context is not considered. (2) Strict application of pattern matching may fail to capture some patterns that contain the proposed patterns. (3) Simple POS tag rules cannot identify some noun phrases that have complex structures using modifiers.

To overcome these limitations, we perform pattern matching by first matching sentences containing lexical connectors, and then extracting the corresponding noun phrases from the text segments either surrounding those connectors or in between them, by analyzing the constituent parse tree structure for the sentences. The idea is that a well-trained parser like the Stanford Parser can be more effective at determining noun phrases than simply matching regular expressions over POS tags.

One challenge in ontology extraction is that noun phrases may be nested in other noun phrases. In this case it is difficult to identify the appropriate noun phrases in the extracted relationship. As an example, in the sentence "Provisions of shading devices, *such as* overhangs or vertical fins." the noun-phrase "shading devices" inside "provision of shading devices" is a nested noun phrase and the correct parent concept of the ISA relationship, not "provision of shading devices." To solve this challenge, we employ a linguistically based heuristic approach that uses hints from an external source, e.g., a general thesaurus like WordNet. A useful cue about the type of a noun phrase can be obtained from its head word. A *head word* is the word that determines the syntactic type of the noun phrase of which it is a member.

Algorithm 1 extracts the best choice for a parent concept given a nested noun phrase (for the parent) and a list of noun phrases (for the child). Lines 4–12 calculate the sum of the similarity between each candidate parent's head word and head words of all children. $MaxSimSum$ indicates the similarity sum for the candidate parent with maximum similarity sum. . The function *Similarity* in line 4 is an invocation of the semantic similarity measure defined in [16]. If two candidates have the same sum, we will choose the shortest one (Line 9–11), because the head word of a parent phrase tends to be closer to the child phrases that specify this parent. When the maximum similarity sum is zero, meaning head words are not found in WordNet (which is possible when we are dealing with a domain-specific corpus), we will try to find the shortest noun phrase, with the head word in plural form if it exists, as a default behavior (Line 13–15).

3.2 SCIP Extension

We can extend the set of pairs derived by generating more SCIPs that exploit the inherent ISA relationship between a complex phrase and its head word and the transitivity of ISA relationship. E.g., consider the SCIP ISA(plumbing equipment, ductile iron pipe). We can extend this by generating the SCIP ISA(equipment, ductile iron pipe). Many existing algorithms make the assumption, that if ISA(NP_1, NP_2), then necessarily ISA(*head*(NP_1), *head*(NP_2)).

We observed on real data sets that this assumption results

Algorithm 1 Parent_NP_Resolution_in_Nested_NP

Input: A nested noun phrase ($NestedNP$) containing the potential parent noun phrase; A set of child noun phrases ($ChildList$).
Output: The appropriate parent noun phrase ($ParentNP$), which is a hypernym of the noun phrases in $ChildList$.

$ParentList \leftarrow$ Recursively extract a list of noun phrases containing the last word in $NestedNP$ from the parse tree
$MaxSimSum = -1$
$CurrentCandidate = null$
for all $Candidate \in ParentList$ **do**
  $SimSum = \sum_{ChildNP \in ChildList} Similarity(headof(Candidate), headof(ChildNP))$
  if $SimSum > MaxSimSum$ **then**
    $CurrentCandidate = Candidate$
    $MaxSimSum = SimSum$
  else if $SimSum == MaxSimSum$ and $length(Candidate) < length(CurrentCandidate)$ **then**
    $CurrentCandidate = Candidate$
  end if
end for
if $MaxSimSum == 0$ **then**
  Return $ParentNP \leftarrow$ shortest $Candidate$ in $ParentList$, breaking ties in favor of a candidate with a plural head word if any, and then arbitrarily.
end if
Return $ParentNP \leftarrow CurrentCandidate$

in many erroneous relationships or trivial relationships that can be found in a general ontology. This is because in many cases, the sense of the head word cannot be disambiguated without modifiers. For example, following this assumption on ISA(points of penetration of the vapor barrier jacket, raw edges) yields ISA(points, edges), which is meaningless!

In summary, we extend every extracted pair ISA(NP_1, NP_2) from a SCIP by generating the additional pair ISA(*head(NP_1)*, *head(NP_2)*). Then we calculate reliability scores for all extracted and extended pairs based on the scoring mechanism described in Section 3.4. Finally, we filter those pairs with scores smaller than average and pick the top ones as seed SCIPs for discovering new patterns in the next iteration.

3.3 Frequent Pattern Discovery

We use the seed ISA/HASA relationships to find new patterns. We adopt a *Frequent-Substring-based Pattern Extraction* approach to achieve this. The idea is to find substrings that frequently occur in between the parent concept and the child concept of a SCIP in the corpus. Using seed instances in the form of SCIP(NP_1, NP_2) as input, and we find co-occurrences of NP_1 and NP_2 in the corpus where the text in between NP_1 and NP_2 is shorter than a pre-defined limit. After collecting text sequences for each SCIP, we find frequent substrings from them. To solve the above problem, we tailor the Generalized Sequential Patterns (GSP) algorithm in [18]. For brevity, we omit the details of this algorithm and refer the reader to the full version of this paper [10].

3.4 Scoring of Patterns and SCIPs

We need a scoring mechanism to select seed SCIPs and seed patterns to identify new patterns and new concept pairs respectively, and decide the stopping criteria for the iterative process. We follow the Point-wise Mutual Information[5] (PMI) framework for scoring patterns and instances. Using the PMI framework, we use the following formulation:

$$pmi(i,p) = \log \frac{\frac{|NP_1,p,NP_2|}{\sum_{\hat{p}\in P',\hat{i}\in I'}|NP_{1_{\hat{i}}},\hat{p},NP_{2_{\hat{i}}}|}}{\frac{|NP_1,*,NP_2|}{\sum_{\hat{i}\in I'}|NP_{1_{\hat{i}}},*,NP_{2_{\hat{i}}}|}\frac{|*,p,*|}{\sum_{\hat{p}inP'}|*,\hat{p},*|}} \quad (1)$$

in which P' is the set of patterns in the current iteration and I' is the set of instances used to find new patterns. In the above equation, we divide the frequency value in the numerator and the denominator with corresponding sum values, namely the sum of co-occurrence frequency for all pairs of instance and pattern, the sum of frequencies of all instances, and the sum of frequencies of all patterns, respectively. Here, $\hat{i}$ ranges over instances, i.e., $\hat{i} = (NP_{1_{\hat{i}}}, NP_{2_{\hat{i}}})$.

In the first iteration of pattern instantiation, we estimate the precision of the initial set of pre-defined patterns by manual validation on a sampled output, and use those estimates as initial scores. The algorithm runs until no more new SCIPs can be found or the average score of patterns produced in the current iteration is smaller than 50% of the average score of patterns from the previous iteration.

4. EXPERIMENTS AND EVALUATION

We evaluated our results on the following three real web datasets: (1) an archive of Architecture, Engineering, and Construction (AEC) scheduling data, design data, meeting notes, and reports used in the construction of a medium sized building, (2) text from Lonely Planet[1] (LP), and (3) medical references from MEDLINE [9] (MED).

We compare LASER with three other algorithms as follows: (1) ESPRESSO [14], (2) GHC [4], and (3) WCL-HG [13]. These algorithms were briefly described in Section 2. LASER and ESPRESSO find ISA/HASA relationships while GHC and WCL-HG are only able to produce ISA relationships.

We would like to measure and compare the *precision* as well as *recall* for the above algorithms. However, given that it is infeasible to fully find all ontological relationships in a large text repository, we measured *relative recall* — the number of valid relationships found by the algorithm divided by the total number of valid relationships found by all algorithms [11]. This allows us to also define *relative F-score* by replacing *recall* with *relative recall*. A more detailed description of the datasets, measures and algorithms is presented in the full version of this paper [10].

4.1 Comparison of ISA Results

Using the stopping criterion in Section 3.4, LASER ran two ISA Pattern Instantiation iterations on all datasets, while ESPRESSO ran one before it reached its stopping criterion.

Table 1 shows the total number of all output ISA relationships for each algorithm and the corresponding precision. We manually validated all relationships produced for AEC. For the other two corpora, we validated random 100 results if there were more than 1,000 relationships, otherwise we did complete validation. LASER1 and LASER2 represent the relationships directly extracted from patterns during iteration 1 and 2; LASER is the total result from all iterations. HW denotes the results containing extended relationships found by the SCIP Extension step (Section 3.2). HW+W represents the result with both head word extension and web extension. Finally, DEF indicates expanded results using web definitions (Merriam-Webster dictionary in this case).

[1] *http://www.lonelyplanet.com*

Table 1: Precision and Total Number of ISA Results

System	AEC		LP		MED	
	Prec	Total	Prec	Total	Prec	Total
LASER1	0.593	617	0.63	2198	0.6	19338
LASER1 HW	0.564	1070	0.5	3995	**0.61**	34390
LASER2	0.453	64	0.644	104	0.37	5323
LASER2 HW	0.459	111	0.642	179	0.42	9674
LASER	0.58	681	0.61	2302	0.55	24661
LASER HW	0.555	1181	0.6	**4174**	0.56	**44064**
ESPRESSO	0.673	55	**0.766**	141	0.53	3472
ESPRESSO HW	**0.674**	95	0.755	229	0.59	5814
ESPRESSO HW+W	0.337/ 0.562	406	0.59/ 0.68	1396	0.43/ 0.5	14077
WCL-HG	0.189	53	0.3	77	0.37	6210
WCL-HG +DEF	0.139/ 0.39	**2353**	0.3/ 0.44	455	0.23/ 0.38	7584
GHC	0.337	734	0.51	1074	0.59	3557

In this scenario, for every definition ⟨DEF⟩ found for the input phrase ⟨TARGET⟩, we add the following to the set of candidate sentences: "⟨TARGET⟩ is a ⟨DEF⟩".

Table 2: Relative Recall and F-score of ISA

System	AEC		LP		MED	
	RR	F	RR	F	RR	F
LASER	0.30	0.40	0.35	0.44	0.39	0.46
LASER HW	**0.49**	**0.52**	**0.62**	**0.61**	**0.71**	**0.63**
ESPRESSO	0.03	0.06	0.03	0.05	0.05	0.09
ESPRESSO HW	0.05	0.09	0.04	0.08	0.10	0.18
ESPRESSO HW+W	0.11	0.17	0.21	0.31	0.18	0.26
WCL-HG	0.01	0.02	0.01	0.02	0.10	0.16
WCL-HG+DEF	0.24	0.17	0.03	0.05	0.07	0.11
GHC	0.19	0.24	0.14	0.22	0.05	0.09

ESPRESSO achieves the best precision on the two smaller datasets and LASER achieves the best precision on MED. Head word extension increases the number of relationships found by both LASER and ESPRESSO, with precision remaining about the same or decreasing a little bit because of errors in finding head words. ESPRESSO's web expansion produces many additional relationships, but it markedly degrades precision.

The two numbers in each precision column of ESPRESSO HW+W and WCL-HG+DEF measure precisions on relationships found in the domain and relationships valid in *any* domain, respectively. For example, ISA(accessories, necklace) is extracted by ESPRESSO on the AEC dataset. This is not valid in the architecture domain because necklace is not a concept in this domain — in this domain, accessories stands for construction or mechanical equipment.

GHC relies heavily on ISA relationships between a term and its head word, e.g., ISA(system, heat recovery system), which are fairly trivial. Neither LASER, nor ESPRESSO output these relationships. The input terms extracted for MED contain a higher percentage of multi-word terms (32%) than those of AEC (23%), so GHC performs much better on the MED corpus: more "trivial" relationships can be found.

The relatively poor performance of WCL-HG compared to the results in the initial publications [12] may be due to the following reasons. (1) Lacking access to a good set of input domain terms, (2) Patterns learned by WCLs over Wikipedia definition sentences are not effective over complex and technical corpuses such as AEC. (3) As noted in [12], in many cases WCLs are only able to match a substring of the complete match phrase. E.g., over AEC, they return

Table 3: Precision and Total Number of HASA Results

System	AEC		LP		MED	
	Prec	Total	Prec	Total	Prec	Total
LASER1	0.415	82	**0.626**	673	**0.42**	4011
LASER2	0	0	0.429	7	0.25	417
LASER	**0.415**	**82**	0.624	**680**	0.39	**4428**
ESPRESSO	0.25	4	0.588	51	0.35	428
ESPRESSO HW	0.11	9	0.521	71	0.31	649
ESPRESSOW HW+W	0.1	10	0.454/ 0.471	121	0.34/ 0.39	1072

ISA(furring, application) as a match, which is too general and is a substring of the correct match ISA(furring, application of thin wood).

Table 2 gives the relative recall and F-score for algorithms on all corpuses. Testing the validity of all relationships from the two larger datasets is impractical, so for those, we estimate *relative recall* by: $RR \approx \frac{precision*|SCIPs|}{\sum precision_x*|SCIPs|_x}$ — the number of valid relationships produced by an algorithm is estimated by the product of sample precision and the number of all generated SCIPs. Summing the estimated valid relationships for all competing algorithms, yields the number of all valid relationships from all systems' output, which is an overestimate of the real value. Therefore, the estimated relative recall is an under estimate but still reflects the difference between systems.

LASER HW outperforms the other algorithms and corresponding extensions in terms of relative recall and F-score, thanks to the large output and stable precision. In contrast, ESPRESSO suffers from low relative recall. Although Espresso's set of SCIPs are valid, the distribution of these seeds in the corpus is unknown beforehand, leading to possibly re-discovering the same pattern repeatedly and hence a consistently low recall. GHC has better relative recall and F-score than ESPRESSO on AEC even when its precision is low on AEC. On the large corpus, GHC has worse relative recall mainly because the agglomerative clustering algorithm does not scale well. As we show later, even running GHC with 5000 terms took more than two days. WCL-HG has its largest relative recall on AEC dataset. The reason is that the algorithm is able to find a large number of web definitions for the input seed target phrases. As a result, it finds almost all its matches from web definitions. However, most of these matches are either not complete or are not relevant in the given domain, which results in a low precision.

4.2 Comparison of HASA Results

LASER ran one HASA Pattern Instantiation iteration on AEC and two iterations on other two datasets. ESPRESSO still ran only one HASA Pattern Instantiation iteration. Table 3 shows that both the precision and number of HASA relationships are worse than ISA relationships for all algorithms. This is because in a corpus, HASA relationships are not as frequent as ISA relationships.

LASER outperforms ESPRESSO in every case over the above datasets. One thing to note is that LASER only extends ISA SCIPs in the SCIP Extension step (Section 3.2), but ESPRESSO extends both ISA and HASA SCIPs. We made this choice because HASA has different semantic meanings from ISA and contains many subtypes [7]. For example, HASA(treatment of occlusive disease, endarterectomy) is a valid relationship from MED, but its head word extension HASA(treatment, endarterectomy) does not make sense be-

Table 4: Relative Recall and F-score of HASA

System	AEC		LP		MED	
	RR	F	RR	F	RR	F
LASER	**0.971**	**0.581**	**0.885**	**0.732**	**0.826**	**0.530**
ESPRESSO	0.029	0.052	0.063	0.114	0.072	0.119
ESPRESSO HW	0.029	0.046	0.077	0.134	0.096	0.147
ESPRESSO HW+W	0.029	0.045	0.115	0.184	0.174	0.230

cause "treatment" is too abstract that "endarterectomy" is not part of "treatment" in the general sense. ESPRESSO's drop in precision when it applies HASA headword extension also reflects this.

The relative recall and F-scores on the three corpora are presented in Table 4. LASER dominates both measurements consistently while ESPRESSO still suffers from low recall.

4.3 Comparison of Running Time

Table 5 shows the running times for extracting ISA relationships. LASER and LASER HW have the same running time because both versions of LASER require headword extension for seed generation. This is also true for ESPRESSO and ESPRESSO HW. LASER is the most efficient algorithm and is between 1.4 times and two orders of magnitude faster than other algorithms. LASER's sequential pattern mining approach is more efficient than ESPRESSO's frequent substring finding using a suffix tree. ESPRESSO HW+W takes even longer because search engines constrain the query rates. The running time for GHC is quadratic in the number of input terms because agglomerative clustering requires pairwise term similarity. Thus, it takes GHC more than two days to finish on an input of 5,000 terms! The bottleneck for WCL-HG is matching each input sentence against all WCLs to find the potential matches. The running time for extracting HASA relationships is similar to the ISA case, and we omit the results for lack of space.

Table 5: ISA extraction running time (in seconds)

System	AEC	LP	MED
LASER	65	107	6,862
ESPRESSO	518	308	9,627
ESPRESSO HW+W	10,939	10,439	72,154
WCL-HG	894	224	41 hours
WCL-HG+DEF	2,710	1043	2 days
GHC	1,160	1,709	>2 days

5. CONCLUSIONS AND FUTURE WORK

Many state-of-the-art algorithms for learning ontologies from free text confine themselves to concepts represented as single-word terms or common compounds. In contrast, we find a richer ontology by covering multi-word terms. We extend previous pattern-based iterative frameworks [8, 14], and make the following contributions: (1) We identify concepts in ISA/HASA relationships by analyzing parse trees instead of simple POS tags, and use an efficient parse-once-use-many-times strategy. (2) We develop a novel algorithm to determine the appropriate noun phrases from nested noun phrases present in the corpus. (3) We tailor sequential pattern mining to find constrained frequent patterns consisting of words, POS tags, and wildcards.

We empirically show on three real web datasets that LASER stably extracts rich and complex concepts and ISA/HASA relationships between them, regardless of the size of corpus or data sparsity. In terms of precision, it is comparable to or better than the competitors while in terms of relative recall and F-score it significantly and consistently outperforms them. Finally, we show that LASER has a significantly better running time compared to the competing algorithms and better scales.

An interesting future challenge is to post-process concepts found by LASER with statistical methods to boost the precision even further while maintaining scalability.

6. ACKNOWLEDGMENTS

This research was supported by the NSERC (Natural Sciences and Engineering Research Council) Business Intelligence Network.

7. REFERENCES

[1] M. Berland and E. Charniak. Finding parts in very large corpora. In *ACL*, pages 57–64. ACL, 1999.
[2] C. Biemann. Ontology learning from text: A survey of methods. *LDV Forum*, 20(2):75–93, 2005.
[3] S. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *ACL*, 1999.
[4] P. Cimiano and S. Staab. Learning concept hierarchies from text with a guided hierarchical clustering algorithm. In *ICML workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*, 2005.
[5] T. Cover and J. Thomas. *Elements of information theory*, volume 6. Wiley Online Library, 1991.
[6] E. Drymonas, K. Zervanou, and E. Petrakis. Unsupervised ontology acquisition from plain texts: the OntoGain system. *Natural Language Processing and Information Systems*, pages 277–287, 2010.
[7] R. Girju, A. Badulescu, and D. Moldovan. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135, 2006.
[8] M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING*. ACL, 1992.
[9] W. Hersh, C. Buckley, T. Leone, and D. Hickam. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *SIGIR*, 1994.
[10] T. Li, P. Chubak, L. V. Lakshmanan, and R. Pottinger. Efficient extraction of ontologies from domain specific text corpora. Technical report, University of British Columbia. http://www.cs.ubc.ca/cgi-bin/tr/2012/TR-2012-04.pdf.
[11] A. Moosavi, T. Li, L. Lakshmanan, and R. Pottinger. Ontectas: Bridging the gap between collaborative tagging systems and structured data. In *CAiSE*, 2011.
[12] R. Navigli and P. Velardi. Learning word-class lattices for definition and hypernym extraction. In *ACL*, pages 1318–1327, 2010.
[13] R. Navigli, P. Velardi, and S. Faralli. A graph-based algorithm for inducing lexical taxonomies from scratch. In *IJCAI*, 2011.
[14] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *COLING*, pages 113–120. ACL, 2006.
[15] P. Pantel and D. Ravichandran. Automatically labeling semantic classes. In *HLT/NAACL*, 2004.
[16] T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet:: Similarity: measuring the relatedness of concepts. In *HLT/NAACL*, pages 38–41, 2004.
[17] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *SIGIR*, pages 206–213, 1999.
[18] Srikant and Agrawal. Mining sequential patterns: Generalizations and performance improve. *EDBT*, 1996.