Feature Selection for Automatic Taxonomy Induction

Hui Yang
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA, 15213
huiyang@cs.cmu.edu

Jamie Callan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA, 15213
callan@cs.cmu.edu

ABSTRACT
Most existing automatic taxonomy induction systems exploit one or more features to induce a taxonomy; nevertheless there is no systematic study examining which are the best features for the task under various conditions. This paper studies the impact of using different features on taxonomy induction for different types of relations and for terms at different abstraction levels. The evaluation shows that different conditions need different technologies or different combination of the technologies. In particular, co-occurrence and lexico-syntactic patterns are good features for *is-a*, *sibling* and *part-of* relations; contextual, co-occurrence, patterns, and syntactic features work well for concrete terms; co-occurrence works well for abstract terms.

Categories and Subject Descriptors
H.3.1 Content Analysis and Indexing.

General Terms
Experimentation, Verification.

Keywords
Ontology Learning, Taxonomy, Semantic Feature.

1. INTRODUCTION
Automatic taxonomy induction is an important task in the fields of Natural Language Processing, Knowledge Management, and Semantic Web. It can be conducted for different types of relations, such as *is-a*, *sibling*, and *part-of*. It can also be conducted for terms with different levels of abstractness, including *concrete terms* and *abstract terms*.

Existing work on automatic taxonomy induction falls into two main categories: *pattern-based* and *clustering-based*. *Pattern-based* approaches [1][3][5] define lexical-syntactic patterns for relations, and use these patterns to discover instances of relations. The approaches are known for their high accuracy in discovering relations. However they cannot find relations which do not explicitly appear in text. *Clustering-based* approaches [6][7] hierarchically cluster terms based on similarities of their meanings usually represented by a vector of features. The approaches complement pattern-based approaches by their ability to discover relations which do not explicitly appear in text. However, they cannot generate relations as accurate as pattern-based approaches.

The common types of features used in clustering-approaches include *contextual*, *co-occurrence*, and *syntactic dependency*. A recent clustering-based approach [7] proposed to incorporate *lexico-syntactic patterns* as one type of features in the clustering-framework, and it is shown to achieve better accuracy for the task.

These heterogeneous features play an important role in automatic taxonomy induction since they represent various technologies in this field. However, there is no systematic study examining which features are the best for the task under various conditions.

This paper presents such a study. In particular, it studies the impact of various features on taxonomy induction for different types of relations and for terms at different abstraction levels.

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.
ACM 978-1-60558-483-6/09/07.

Table 1: Lexico-Syntactic Patterns.

Hypernym Patterns	*Sibling Patterns*
NP_x (,)?and/or other NP_y	NP_x and/or NP_y
such NP_y as NP_x	*Part-of Patterns*
NP_y (,)? such as NP_x	NP_x of NP_y
NP_y (,)? including NP_x	NP_y's NP_x
NP_y (,)? especially NP_x	NP_y has/had/have NP_x
NP_y like NP_x	NP_y is made (up)? of NP_x
NP_y called NP_x	NP_y comprises NP_x
NP_x is a/an NP_y	NP_y consists of NP_x
NP_x , a/an NP_y	

2. FEATURES
The features used in this work are indicators of semantic relations between terms. Given two input terms (c_x,c_y), a feature is defined as a function generating a single numeric score $h(c_x,c_y)\in\mathbb{R}$ or a vector of numeric scores $h(c_x,c_y)\in\mathbb{R}^n$. We study five sets of features, including *contextual*, *co-occurrence*, *syntactic dependency*, *lexical-syntactic patterns*, and *miscellaneous*.

The first set of features captures *contextual* information about terms. Based on the Distributional Hypothesis [4], we develop the following features: (1) *Global Context KL-Divergence*: The global context of each input term is the search results collected through querying search engines against the auxiliary corpora. The global context is built into a language model for each term. This feature function measures the Kullback-Leibler divergence (KL divergence) between the language models associated with two inputs. (2) *Local Context KL-Divergence*: The local context is collected by extracting the left two and the right two words surrounding an input term. Similarly to the global context, the local context is built into a language model for each term; the feature function outputs KL divergence between the models.

The second set of features is *co-occurrence*. We measure co-occurrence by point-wise mutual information between two terms:

$$pmi(c_x,c_y)=\log\frac{Count(c_x,c_y)}{Count(c_x)Count(c_y)}$$

where *Count(.)* is defined as the number of documents or sentences containing the term(s); or *n* as in "Results 1-10 of about *n* for *term*" appearing on the first page of Google search results for a term or the concatenation of a term pair. Based on different definitions of *Count(.)*, we have (3) *Document PMI*, (4) *Sentence PMI*, and (5) *Google PMI* as co-occurrence features.

The third set of features employs *syntactic dependency* analysis. We use (6) *Minipar Syntactic Distance* to measure the average length of the shortest syntactic paths (in the first syntactic parse tree returned by Minipar[1]) between two terms over sentences containing them; (7) *Modifier Overlap*, (8) *Object Overlap*, (9) *Subject Overlap*, and (10) *Verb Overlap* to measure the number of overlaps between modifiers, objects, subjects, and verbs, respectively, for the two input terms in sentences containing them.

[1] http://www.cs.ualberta.ca/lindek/minipar.htm.

The fourth set of feature is *lexical-syntactic patterns*. We use (11) *Hypernym Patterns* proposed by [1] and [5], (12) *Sibling Patterns* which are basically conjunctions, and (13) *Part-of Patterns* proposed by [1] and [3]. Each feature function returns a vector of scores for the two input terms, one score per pattern. A score is 1 if the terms appear with that pattern in text, 0 otherwise. Table 1 lists all the patterns used in this work.

The last set of features is *miscellaneous*. We use (14) *Word Length Difference* to measure the length difference between two terms, and (15) *Definition Overlap* to measure the word overlaps between term definitions by querying Google with "define:*term*".

3. EXPERIMENTS

The gold standards used in the evaluation are 50 hypernym taxonomies from WordNet [2] and 50 from ODP (Open Directory Project), and 50 meronym taxonomies from WordNet. In WordNet taxonomies, we use the word senses within a particular taxonomy to eliminate ambiguity. In ODP taxonomies, we parse the topic lines, such as "Topic r:id=`Top/Arts/Movies'", in the XML databases to obtain relations such as *is_a(movies, arts)*.

We also use two auxiliary datasets: *Wikipedia corpus* and *Google Corpus*. *Wikipedia corpus* is the entire Wikipedia corpus downloaded and indexed by Indri. *Google corpus* is a collection of the top 1000 Google documents obtained by querying Google using each term, and each term pair. In particular, both corpora are split into sentences and used to generate contextual, co-occurrence, syntactic dependency and pattern features.

We evaluate the quality of automatically generated taxonomies by comparing them with the gold standards in terms of F1-measure for the relations. Leave-one-out cross validation is used to average the system performance over different training and testing datasets; the averaged F1-measure is reported across 50 runs.

3.1 Features vs. Relations

This section studies the effect of using 15 heterogeneous features (grouped in 5 categories) for different types of relations. Each category is utilized one by one. Table 2 shows the F1-measure of using various features on automatic taxonomy induction on WordNet datasets for *is-a, sibling*, and *part-of* relations. Bold font indicates that good performance in a column.

Table 2 shows that co-occurrence and lexico-syntactic patterns work equally well and significantly improve taxonomy induction for all three types of relations. Contextual features work well for identifying sibling relations, but not for *is-a* and *part-of*. Syntactic features show the similar results as contextual features because four out of five syntactic features, (*Modifier Overlap, Subject Overlap, Object Overlap, and Verb overlap*) are surrounding context to a term. The row of *"All"* shows the F1-measure when combining all the features for the task, and it consistently achieves the best performance for all the three relations.

3.2 Features vs. Abstractness

This section studies the impact of different feature categories on terms at different abstraction levels. The F1-measure is evaluated for terms at each level of a taxonomy, not the whole taxonomy. Tables 3 and 4 demonstrate the F1-measure of using each feature category alone on each abstraction level. Columns 2-6 are indices of the levels in a taxonomy. The larger the indices are, the lower the levels. Higher levels contain abstract terms, while lower levels contain concrete terms. L_1 is ignored here since it only contains the root. Bold font indicates good performance in a column.

Both tables show that abstract terms and concrete terms favor different sets of features. In particular, contextual, co-occurrence, pattern, and syntactic features work well for terms at L_4-L_6, i.e., the concrete terms; co-occurrence works well for terms at L_2-L_3, i.e., the abstract terms.

Table 2: F1-measure for Features vs. Relations: WordNet.

Feature Type	*is-a*	*sibling*	*part-of*	***Benefited Relations***
Contextual	0.21	**0.42**	0.12	*sibling*
Co-occur.	**0.48**	**0.41**	**0.28**	*All*
Patterns	**0.46**	**0.41**	**0.30**	*All*
Syntactic	0.22	**0.36**	0.12	*sibling*
Misc.	0.14	0.17	0.12	
All	***0.82***	***0.79***	***0.61***	***All***
Best Features	Co-occur., patterns	Contextual, co-occur., patterns	Co-occur., patterns	

Table 3: F1-measure for Features vs. Abstractness: WordNet/*is-a*.

Feature Type	L_2	L_3	L_4	L_5	L_6
Contextual	0.29	0.31	**0.35**	**0.36**	**0.36**
Co-occurrence	**0.47**	**0.56**	**0.45**	**0.41**	**0.41**
Patterns	**0.47**	**0.44**	**0.42**	**0.39**	**0.40**
Syntactic	0.31	0.28	**0.36**	**0.38**	**0.39**
Misc.	0.14	0.14	0.14	0.14	0.14

Table 4: F1-measure for Features vs. Abstractness: ODP/*is-a*.

Feature Type	L_2	L_3	L_4	L_5	L_6
Contextual	**0.30**	**0.30**	**0.33**	**0.29**	**0.29**
Co-occurrence	**0.34**	**0.36**	**0.34**	**0.31**	**0.31**
Patterns	0.23	0.25	**0.30**	**0.28**	**0.28**
Syntactic	0.18	0.18	0.23	**0.27**	**0.27**
Misc.	0.14	0.14	0.14	0.13	0.13

We also observe that for abstract terms in WordNet, patterns work better than contextual features; while for abstract terms in ODP, the conclusion is the opposite. This may be because WordNet has a rigid definition of hypernyms, and hence it favors lexico-syntactic patterns which require more rigidity. While ODP contains more noise, and hence it favors features requiring less rigidity, such as the contextual features generated from the Web.

4. CONCLUSIONS

This paper studies the impact of various features on automatic taxonomy induction for different types of relations and for terms at different abstraction levels. The experiments show that co-occurrence and lexico-syntactic patterns are good features for common relations, such as *is-a, sibling,* and *part-of*. Contextual and syntactic features are only good for *sibling* relations. Moreover, the experiments show that abstract terms and concrete terms favor different sets of features. Contextual, co-occurrence, patterns, and syntactic features work well for concrete terms; co-occurrence works well for abstract terms.

5. REFERENCES

[1] P. Cimiano and J. Wenderoth. 2007. Automatic Acquisition of Ranked Qualia Structures from the Web. ACL'07.

[2] C. Fellbuam. WordNet: An Electronic Lexical Database. MIT Press.1998.

[3] R. Girju, A. Badulescu, and D. Moldovan. 2003. Learning Semantic Constraints for the Automatic Discovery of Part-Whole Relations. HLT'03.

[4] Z. Harris. 1985. Distributional Structure. In: J. J. Katz (ed.), The Philosophy of Linguistics. Oxford University Press.

[5] M. Hearst, 1992. Automatic acquisition of hyponyms from large text corpora. COLING'92.

[6] P. Pantel and D Lin, 2002. Discovering word senses from text. SIGKDD'02.

[7] H. Yang and J. Callan. 2009. A Metric-based Framework for Automatic Taxonomy Induction. ACL'09.