Towards Terascale Knowledge Acquisition

Patrick Pantel, **Deepak Ravichandran** and **Eduard Hovy**
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
{pantel,ravichan,hovy}@isi.edu

Abstract

Although vast amounts of textual data are freely available, many NLP algorithms exploit only a minute percentage of it. In this paper, we study the challenges of working at the terascale. We present an algorithm, designed for the terascale, for mining *is-a* relations that achieves similar performance to a state-of-the-art linguistically-rich method. We focus on the accuracy of these two systems as a function of processing time and corpus size.

1 Introduction

The Natural Language Processing (NLP) community has recently seen a growth in corpus-based methods. Algorithms light in linguistic theories but rich in available training data have been successfully applied to several applications such as machine translation (Och and Ney 2002), information extraction (Etzioni et al. 2004), and question answering (Brill et al. 2001).

In the last decade, we have seen an explosion in the amount of available digital text resources. It is estimated that the Internet contains hundreds of terabytes of text data, most of which is in an unstructured format. Yet, many NLP algorithms tap into only megabytes or gigabytes of this information.

In this paper, we make a step towards acquiring semantic knowledge from terabytes of data. We present an algorithm for extracting *is-a* relations, designed for the terascale, and compare it to a state of the art method that employs deep analysis of text (Pantel and Ravichandran 2004). We show that by simply utilizing more data on this task, we can achieve similar performance to a linguistically-rich approach. The current state of the art co-occurrence model requires an estimated 10 years just to parse a 1TB corpus (see Table 1). Instead of using a syntactically motivated co-occurrence approach as above, our system uses lexico-syntactic rules. In particular, it finds lexico-POS patterns by making modifications to the basic edit distance algorithm. Once these patterns have been learnt, the algorithm for finding new *is-a* relations runs in $O(n)$, where n is the number of sentences.

In semantic hierarchies such as WordNet (Miller 1990), an *is-a* relation between two words x and y represents a subordinate relationship (i.e. x is more specific than y). Many algorithms have recently been proposed to automatically mine *is-a* (hyponym/hypernym) relations between words. Here, we focus on *is-a* relations that are characterized by the questions "*What/Who is X*?" For example, Table 2 shows a sample of 10 *is-a* relations discovered by the algorithms presented in this paper. In this table, we call *azalea*, *tiramisu*, and *Winona Ryder* instances of the respective concepts *flower*, *dessert* and *actress*. These kinds of *is-a* relations would be useful for various purposes such as ontology construction, semantic information retrieval, question answering, etc.

The main contribution of this paper is a comparison of the quality of our pattern-based and co-occurrence models as a function of processing time and corpus size. Also, the paper lays a foundation for terascale acquisition of knowledge. We will show that, for very small or very large corpora or for situations where recall is valued over precision, the pattern-based approach is best.

2 Relevant Work

Previous approaches to extracting *is-a* relations fall under two categories: pattern-based and co-occurrence-based approaches.

2.1 Pattern-based approaches

Marti Hearst (1992) was the first to use a pattern-based approach to extract hyponym relations from a raw corpus. She used an iterative process to semi-automatically learn patterns. However, a corpus of 20MB words yielded only 400 examples. Our pattern-based algorithm is very similar to the one used by Hearst. She uses seed examples to manually discover her patterns whearas we use a minimal edit distance algorithm to automatically discover the patterns.

Table 1. Approximate processing time on a single Pentium-4 2.5 GHz machine.

TOOL	*15 GB ORPUS*	*1 TB CORPUS*
POS Tagger	2 days	125 days
NP Chunker	3 days	214 days
Dependency Parser	56 days	10.2 years
Syntactic Parser	5.8 years	388.4 years

Riloff and Shepherd (1997) used a semi-automatic method for discovering similar words using a few seed examples by using pattern-based techniques and human supervision. Berland and Charniak (1999) used similar pattern-based techniques and other heuristics to extract meronymy (part-whole) relations. They reported an accuracy of about 55% precision on a corpus of 100,000 words. Girju et al. (2003) improved upon Berland and Charniak's work using a machine learning filter. Mann (2002) and Fleischman et al. (2003) used part of speech patterns to extract a subset of hyponym relations involving proper nouns.

Our pattern-based algorithm differs from these approaches in two ways. We learn lexico-POS patterns in an automatic way. Also, the patterns are learned with the specific goal of scaling to the terascale (see Table 2).

2.2 Co-occurrence-based approaches

The second class of algorithms uses co-occurrence statistics (Hindle 1990, Lin 1998). These systems mostly employ clustering algorithms to group words according to their meanings in text. Assuming the distributional hypothesis (Harris 1985), words that occur in similar grammatical contexts are similar in meaning. Curran and Moens (2002) experimented with corpus size and complexity of proximity features in building automatic thesauri. CBC (Clustering by Committee) proposed by Pantel and Lin (2002) achieves high recall and precision in generating similarity lists of words discriminated by their meaning and senses. However, such clustering algorithms fail to name their classes.

Caraballo (1999) was the first to use clustering for labeling *is-a* relations using conjunction and apposition features to build noun clusters. Recently, Pantel and Ravichandran (2004) extended this approach by making use of all syntactic dependency features for each noun.

3 Syntactical co-occurrence approach

Much of the research discussed above takes a similar approach of searching text for simple surface or lexico-syntactic patterns in a bottom-up approach. Our co-occurrence model (Pantel and Ravichandran 2004) makes use of semantic classes like those generated by CBC. Hyponyms are generated in a top-down approach by naming each group of words and assigning that name as a hyponym of each word in the group (i.e., one hyponym per instance/group label pair).

Table 2. Sample of 10 i*s-a* relationships discovered by our co-occurrence and pattern-based systems.

CO-OCCURRENCE SYSTEM		*PATTERN-BASED SYSTEM*	
Word	Hypernym	Word	Hypernym
azalea	flower	American	airline
bipolar disorder	disease	Bobby Bonds	coach
Bordeaux	wine	radiation therapy	cancer treatment
Flintstones	television show	tiramisu	dessert
salmon	fish	Winona Ryder	actress

The input to the extraction algorithm is a list of semantic classes, in the form of clusters of words, which may be generated from any source. For example, following are two semantic classes discovered by CBC:

(A) peach, pear, pineapple, apricot, mango, raspberry, lemon, cherry, strawberry, melon, blueberry, fig, apple, plum, nectarine, avocado, grapefruit, papaya, banana, cantaloupe, cranberry, blackberry, lime, orange, tangerine, ...

(B) Phil Donahue, Pat Sajak, Arsenio Hall, Geraldo Rivera, Don Imus, Larry King, David Letterman, Conan O'Brien, Rosie O'Donnell, Jenny Jones, Sally Jessy Raphael, Oprah Winfrey, Jerry Springer, Howard Stern, Jay Leno, Johnny Carson, ...

The extraction algorithm first labels concepts (A) and (B) with *fruit* and *host* respectively. Then, *is-a* relationships are extracted, such as: *apple is a fruit*, *pear is a fruit*, and *David Letterman is a host*. An instance such as *pear* is assigned a hypernym *fruit* not because it necessarily occurs in any particular syntactic relationship with the word *fruit*, but because it belongs to the class of instances that does. The labeling of semantic classes is performed in three phases, as outlined below.

3.1 Phase I

In the first phase of the algorithm, feature vectors are extracted for each word that occurs in a semantic class. Each feature corresponds to a grammatical context in which the word occurs. For example, "catch __" is a verb-object context. If the word *wave* occurred in this context, then the context is a feature of *wave*.

We then construct a mutual information vector $MI(e) = (mi_{e1}, mi_{e2}, \ldots, mi_{em})$ for each word e, where mi_{ef} is the pointwise mutual information between word e and context f, which is defined as:

$$mi_{ef} = \log \frac{\frac{c_{ef}}{N}}{\frac{\sum_{i=1}^{n} c_{if}}{N} \times \frac{\sum_{j=1}^{m} c_{ej}}{N}}$$

{Phil Donahue,Pat Sajak,Arsenio Hall}		
N:gen:N		
talk show	93	11.77
television show	24	11.30
TV show	25	10.45
show	255	9.98
audience	23	7.80
joke	5	7.37
V:subj:N		
joke	39	7.11
tape	10	7.09
poke	15	6.87
host	40	6.47
co-host	4	6.14
banter	3	6.00
interview	20	5.89
N:appo:N		
host	127	12.46
comedian	12	11.02
King	13	9.49
star	6	7.47

Figure 1. Excerpt of the grammatical signature for the *television host* class.

where n is the number of elements to be clustered, c_{ef} is the frequency count of word e in grammatical context f, and N is the total frequency count of all features of all words.

3.2 Phase II

Following (Pantel and Lin 2002), a committee for each semantic class is constructed. A committee is a set of representative elements that unambiguously describe the members of a possible class. For example, in one of our experiments, the committees for semantic classes (A) and (B) from Section 3 were:

A) peach, pear, pineapple, apricot, mango, raspberry, lemon, blueberry

B) Phil Donahue, Pat Sajak, Arsenio Hall, Geraldo Rivera, Don Imus, Larry King, David Letterman

3.3 Phase III

By averaging the feature vectors of the committee members of a particular semantic class, we obtain a grammatical template, or signature, for that class. For example, Figure 1 shows an excerpt of the grammatical signature for semantic class (B). The vector is obtained by averaging the feature vectors of the words in the committee of this class. The "*V:subj:N:joke*" feature indicates a subject-verb relationship between the class and the verb *joke* while "*N:appo:N:host*" indicates an apposition relationship between the class and the noun *host*. The two columns of numbers indicate the frequency and mutual information scores.

To name a class, we search its signature for certain relationships known to identify class labels. These relationships, automatically learned in (Pantel and Ravichandran 2004), include appositions, nominal subjects, such as relationships, and like relationships. We sum up the mutual information scores for each term that occurs in these relationships with a committee of a class. The highest scoring term is the name of the class.

The syntactical co-occurrence approach has worst-case time complexity $O(n^2k)$, where n is the number of words in the corpus and k is the feature-space (Pantel and Ravichandran 2004). Just to parse a 1 TB corpus, this approach requires approximately 10.2 years (see Table 2).

4 Scalable pattern-based approach

We propose an algorithm for learning highly scalable lexico-POS patterns. Given two sentences with their surface form and part of speech tags, the algorithm finds the optimal lexico-POS alignment. For example, consider the following 2 sentences:

1) Platinum is a precious metal.
2) Molybdenum is a metal.

Applying a POS tagger (Brill 1995) gives the following output:

Surface	*Platinum*	*is*	*a*	*precious*	*metal*	.
POS	*NNP*	*VBZ*	*DT*	*JJ*	*NN*	.

Surface	*Molybdenum*	*is*	*a*	*metal*	.
POS	*NNP*	*VBZ*	*DT*	*NN*	.

A very good pattern to generalize from the alignment of these two strings would be

Surface		*is*	*a*		*metal*	.
POS	*NNP*					.

We use the following notation to denote this alignment: "*_NNP is a (*s*) metal.*", where "*_NNP represents the POS tag NNP*".

To perform such alignments we introduce two wildcard operators, skip (*s*) and wildcard (*g*). The skip operator represents 0 or 1 instance of any word (similar to the \w* pattern in Perl), while the wildcard operator represents exactly 1 instance of any word (similar to the \w+ pattern in Perl).

4.1 Algorithm

We present an algorithm for learning patterns at multiple levels. Multilevel representation is defined as the different levels of a sentence such as the lexical level and POS level. Consider two strings $a(1, n)$ and $b(1, m)$ of lengths n and m respectively. Let $a_1(1, n)$ and $a_2(1, n)$ be the level 1 (lexical level) and level 2 (POS level) representations for the string $a(1, n)$. Similarly, let $b_1(1, m)$ and $b_2(1, m)$ be the level 1 and level 2 representations for the string $b(1, m)$. The algorithm consists of two parts: calculation of the minimal edit distance and retrieval of an optimal pattern. The minimal edit distance algorithm calculates the number of edit operations (insertions, deletions and replacements) required to change one string to another string. The optimal pattern is retrieved by

Table 3. Top 15 lexico-syntactic patterns discovered by our system.

X, or Y	X, _DT Y _(WDT|IN)	Y like X and
X, (a|an) Y	X, _RB known as Y	_NN, X and other Y
X, Y	X (Y)	Y, including X,
Y, or X	Y such as X	Y, such as X
X is a Y	X, _RB called Y	Y, especially X

keeping track of the edit operations (which is the second part of the algorithm).

Algorithm for calculating the minimal edit distance between two strings

```
D[0,0]=0
for i = 1 to n do  D[i,0] = D[i-1,0] + cost(insertion)
for j = 1 to m do D[0,j] = D[0,j-1] + cost(deletion)
for i = 1 to n do
   for j = 1 to m do
      D[i,j] = min(  D[i-1,j-1] + cost(substitution),
                     D[i-1,j] + cost(insertion),
                     D[i,j-1] + cost(deletion))
Print (D[n,m])
```

Algorithm for optimal pattern retrieval

```
i = n, j = m;
while i ≠ 0 and j ≠ 0
   if D[i,j] = D[i-1,j] + cost(insertion)
      print (*s*), i = i-1
   else if D[i,j] = D[i,j-1] + cost(deletion)
      print(*s*), j = j-1
   else if a1i = b1j
      print (a1i), i = i -1, j = j =1
   else if a2i = b2j
      print (a2i), i = i -1, j = j =1
   else
      print (*g*), i = i -1, j = j =1
```

We experimentally set (by trial and error):

$cost(\text{insertion}) = 3$

$cost(\text{deletion}) = 3$

$cost(\text{substitution}) = 0$ if $a_{1i}=b_{1j}$

$= 1$ if $a_{1i}\neq b_{1j}$, $a_{2i}=b_{2j}$

$= 2$ if $a_{1i}\neq b_{1j}$, $a_{2i}\neq b_{2j}$

4.2 Implementation and filtering

The above algorithm takes $O(y^2)$ time for every pair of strings of length at most y. Hence, if there are x strings in the collection, each string having at most length y, the algorithm has time complexity $O(x^2y^2)$ to extract all the patterns in the collection.

Applying the above algorithm on a corpus of 3GB with 50 *is-a* relationship seeds, we obtain a set of 600 lexico-POS. Following are two of them:

1) X_JJ#NN|JJ#NN#NN|NN _CC Y_JJ#JJ#NN|JJ|NNS|NN|JJ#NNS|NN#NN|JJ#NN|JJ#NN#NN
   e.g. …***caldera*** or ***lava lake***…

2) X_NNP#NNP|NNP#NNP#NNP#NNP#NNP#CC#NNP|NNP|VBN|NN#NN|VBG#NN|NN ,_, _DT Y_NN#IN#NN|JJ#JJ#NN|JJ|NN|NN#IN#NNP|NNP#NNP|NN#NN|JJ#NN|JJ#NN#NN
   e.g. …***leukemia***, the ***cancer*** of ...

Note that we store different POS variations of the anchors *X* and *Y*. As shown in example 1, the POS variations of the anchor *X* are (JJ NN, JJ NN NN, NN). The variations for anchor *Y* are (JJ JJ NN, JJ, etc.). The reason is quite straightforward: we need to determine the boundary of the anchors *X* and *Y* and a reasonable way to delimit them would be to use POS information. All the patterns produced by the multi-level pattern learning algorithm were generated from positive examples. From amongst these patterns, we need to find the most important ones. This is a critical step because frequently occurring patterns have low precision whereas rarely occurring patterns have high precision. From the Information Extraction point of view neither of these patterns is very useful. We need to find patterns with relatively *high occurrence* and *high precision*. We apply the log likelihood principle (Dunning 1993) to compute this score. The top 15 patterns according to this metric are listed in Table 3 (we omit the POS variations for visibility). Some of these patterns are similar to the ones discovered by Hearst (1992) while other patterns are similar to the ones used by Fleischman et al. (2003).

4.3 Time complexity

To extract hyponym relations, we use a fixed number of patterns across a corpus. Since we treat each sentences independently from others, the algorithm runs in linear time O(n) over the corpus size, where n is number of sentences in the corpus.

5 Experimental Results

In this section, we empirically compare the pattern-based and co-occurrence-based models presented in Section 3 and Section 4. The focus is on the precision and recall of the systems as a function of the corpus size.

5.1 Experimental Setup

We use a 15GB newspaper corpus consisting of TREC9, TREC 2002, Yahoo! News ~0.5GB, AP newswire ~2GB, New York Times ~2GB, Reuters ~0.8GB, Wall Street Journal ~1.2GB, and various online news website ~1.5GB. For our experiments, we extract from this corpus six data sets of different sizes: 1.5MB, 15 MB, 150 MB, 1.5GB, 6GB and 15GB.

For the co-occurrence model, we used Minipar (Lin 1994), a broad coverage parser, to parse each data set. We collected the frequency counts of the grammatical relationships (contexts) output by Minipar and used them to compute the pointwise mutual information vectors described in Section 3.1. For the pattern-based approach, we use Brill's POS tagger (1995) to tag each data set.

5.2 Precision

We performed a manual evaluation to estimate the precision of both systems on each dataset. For each dataset, both systems extracted a set of *is-a*

Table 4. *Is-a* relationships assigned to three randomly selected words (using pattern-based system on 15GB dataset).

Random Word	*Human*	*WordNet*	*Pattern-based System (Ranked)*
Sanwa Bank	bank	none	subsidiary / lender / bank
MCI Worldcom Inc.	telecommunications company	none	phone company / competitor / company
cappuccino	beverage	none	item / food / beverage

Table 5. Average precision, top-3 precision, and MRR for both systems on each dataset.

	Pattern System			*Co-occurrence System*		
	Prec	*Top-3*	*MRR*	*Prec*	*Top-3*	*MRR*
1.5MB	38.7%	41.0%	41.0%	4.3%	8.0%	7.3%
15MB	39.1%	43.0%	41.5%	14.6%	32.0%	24.3%
150MB	40.6%	46.0%	45.5%	51.1%	73.0%	67.0%
1.5GB	40.4%	39.0%	39.0%	56.7%	88.0%	77.7%
6GB	46.3%	52.0%	49.7%	64.9%	90.0%	78.8%
15GB	55.9%	54.0%	52.0%	Too large to process		

Table 6. Lenient average precision, top-3 precision, and MRR for both systems on each dataset.

	Pattern System			*Co-occurrence System*		
	Prec	*Top-3*	*MRR*	*Prec*	*Top-3*	*MRR*
1.5MB	56.6%	60.0%	60.0%	12.4%	20.0%	15.2%
15MB	57.3%	63.0%	61.0%	23.2%	50.0%	37.3%
150MB	50.7%	56.0%	55.0%	60.6%	78.0%	73.2%
1.5GB	52.6%	51.0%	51.0%	69.7%	93.0%	85.8%
6GB	61.8%	69.0%	67.5%	78.7%	92.0%	86.2%
15GB	67.8%	67.0%	65.0%	Too large to process		

relationships. Six sets were extracted for the pattern-based approach and five sets for the co-occurrence approach (the 15GB corpus was too large to process using the co-occurrence model – see dependency parsing time estimates in Table 2).

From each resulting set, we then randomly selected 50 words along with their top 3 highest ranking *is-a* relationships. For example, Table 4 shows three randomly selected names for the pattern-based system on the 15GB dataset. For each word, we added to the list of hypernyms a human generated hypernym (obtained from an annotator looking at the word without any system or WordNet hyponym). We also appended the WordNet hypernyms for each word (only for the top 3 senses). Each of the 11 random samples contained a maximum of 350 *is-a* relationships to manually evaluate (50 random words with top 3 system, top 3 WordNet, and human generated relationship).

We presented each of the 11 random samples to two human judges. The 50 randomly selected words, together with the system, human, and WordNet generated *is-a* relationships, were randomly ordered. That way, there was no way for a judge to know the source of a relationship nor each system's ranking of the relationships. For each relationship, we asked the judges to assign a score of *correct*, *partially correct*, or *incorrect*. We then computed the average precision of the system, human, and WordNet on each dataset. We also computed the percentage of times a correct relationship was found in the top 3 *is-a* relationships of a word and the mean reciprocal rank (MRR). For each word, a system receives an MRR score of 1 / *M*, where *M* is the rank of the first name judged correct. Table 5 shows the results comparing the two automatic systems. Table 6 shows similar results for a more lenient evaluation where both *correct* and *partially correct* are judged correct.

For small datasets (below 150MB), the pattern-based method achieves higher precision since the co-occurrence method requires a certain critical mass of statistics before it can extract useful class signatures (see Section 3). On the other hand, the pattern-based approach has relatively constant precision since most of the *is-a* relationships selected by it are fired by a single pattern. Once the co-occurrence system reaches its critical mass (at 150MB), it generates much more precise hyponyms. The Kappa statistics for our experiments were all in the range 0.78 – 0.85.

Table 7 and Table 8 compare the precision of the pattern-based and co-occurrence-based methods with the human and WordNet hyponyms. The variation between the human and WordNet scores across both systems is mostly due to the relative cleanliness of the tokens in the co-occurrence-based system (due to the parser used in the approach). WordNet consistently generated higher precision relationships although both algorithms approach WordNet quality on 6GB (the pattern-based algorithm even surpasses WordNet precision on 15GB). Furthermore, WordNet only generated a hyponym 40% of the time. This is mostly due to the lack of proper noun coverage in WordNet.

On the 6 GB corpus, the co-occurrence approach took approximately 47 single Pentium-4 2.5 GHz processor days to complete, whereas it took the pattern-based approach only four days to complete on 6 GB and 10 days on 15 GB.

5.3 Recall

The co-occurrence model has higher precision than the pattern-based algorithm on most datasets.

Table 7. Average precision of the pattern-based system vs. WordNet and human hyponyms.

	PRECISION			MRR		
	Pat.	*WNet*	*Human*	*Pat.*	*WNet*	*Human*
1.5MB	38.7%	45.8%	83.0%	41.0%	84.4%	83.0%
15MB	39.1%	52.4%	81.0%	41.5%	95.0%	91.0%
150MB	40.6%	49.4%	84.0%	45.5%	88.9%	94.0%
1.5GB	40.4%	43.4%	79.0%	39.0%	93.3%	89.0%
6GB	46.3%	46.5%	76.0%	49.7%	75.0%	76.0%
15GB	55.9%	45.6%	79.0%	52.0%	78.0%	79.0%

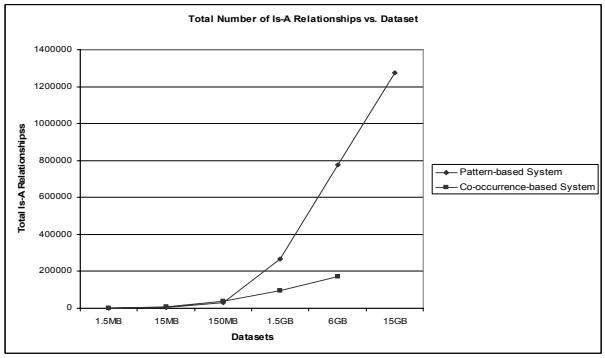

Figure 2. Number of *is-a* relationships extracted by the pattern-based and co-occurrence-based approaches.

Table 8. Average precision of the co-occurrence-based system vs. WordNet and human hyponyms.

	PRECISION			MRR		
	Co-occ	*WNet*	*Human*	*Co-occ*	*WNet*	*Human*
1.5MB	4.3%	42.7%	52.7%	7.3%	87.7%	95.0%
15MB	14.6%	38.1%	48.7%	24.3%	86.6%	95.0%
150MB	51.1%	57.5%	65.8%	67.0%	85.1%	98.0%
1.5GB	56.7%	62.8%	70.3%	77.7%	93.0%	98.0%
6GB	64.9%	68.9%	75.2%	78.8%	94.3%	98.0%

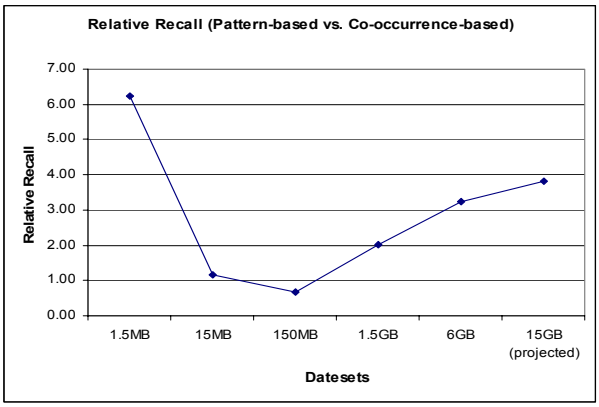

Figure 3. Relative recall of the pattern-based approach relative to the co-occurrence approach.

However, Figure 2 shows that the pattern-based approach extracts many more relationships.

Semantic extraction tasks are notoriously difficult to evaluate for recall. To approximate recall, we defined a relative recall measure and conducted a question answering (QA) task of answering definition questions.

5.3.1 Relative recall

Although it is impossible to know the number of *is-a* relationships in any non-trivial corpus, it is possible to compute the recall of a system relative to another system's recall. The recall of a system A, R_A, is given by the following formula:

$$R_A = \frac{C_A}{C}$$

where C_A is the number of correct *is-a* relationships extracted by A and C is the total number of correct *is-a* relationships in the corpus. We define relative recall of system A given system B, $R_{A,B}$, as:

$$R_{A,B} = \frac{R_A}{R_B} = \frac{C_A}{C_B}$$

Using the precision estimates, P_A, from the previous section, we can estimate $C_A \approx P_A \times |A|$, where A is the total number of *is-a* relationships discovered by system A. Hence,

$$R_{A,B} = \frac{P_A \times |A|}{P_B \times |B|}$$

Figure 3 shows the relative recall of A = pattern-based approach relative to B = co-occurrence model. Because of sparse data, the pattern-based approach has much higher precision and recall (six times) than the co-occurrence approach on the small 15MB dataset. In fact, only on the 150MB dataset did the co-occurrence system have higher recall. With datasets larger than 150MB, the co-occurrence algorithm reduces its running time by filtering out grammatical relationships for words that occurred fewer than k = 40 times and hence recall is affected (in contrast, the pattern-based approach may generate a hyponym for a word that it only sees once).

5.3.2 Definition questions

Following Fleischman et al. (2003), we select the 50 definition questions from the TREC2003 (Voorhees 2003) question set. These questions are of the form "Who is X?" and "What is X?" For each question (e.g., "Who is Niels Bohr?", "What is feng shui?") we extract its respective instance (e.g., "Neils Bohr" and "feng shui"), look up their corresponding hyponyms from our *is-a* table, and present the corresponding hyponym as the answer. We compare the results of both our systems with WordNet. We extract at most the top 5 hyponyms provided by each system. We manually evaluate the three systems and assign 3 classes "Correct (C)", "Partially Correct (P)" or "Incorrect (I)" to each answer.

This evaluation is different from the evaluation performed by the TREC organizers for definition questions. However, by being consistent across all

Table 9. QA definitional evaluations for pattern-based system.

	TOP-1		TOP5	
	Strict	*Lenient*	*Strict*	*Lenient*
1.5MB	0%	0%	0%	0%
15MB	0%	0%	0%	0%
150MB	2.0%	2.0%	2.0%	2.0%
1.5GB	16.0%	22.0%	20.0%	22.0%
6GB	38.0%	52.0%	56.0%	62.0%
15GB	38.0%	52.0%	70.0%	74.0%

systems during the process, these evaluations give an indication of the recall of the knowledge base. We measure the performance on the top 1 and the top 5 answers returned by each system. Table 9 and Table 10 show the results.

The corresponding scores for WordNet are 38% accuracy in both the top-1 and top-5 categories (for both strict and lenient). As seen in this experiment, the results for both the pattern-based and co-occurrence-based systems report very poor performance for data sets up to 150 MB. However, there is an increase in performance for both systems on the 1.5 GB and larger datasets. The performance of the system in the top 5 category is much better than that of WordNet (38%). There is promise for increasing our system accuracy by re-ranking the outputs of the top-5 hypernyms.

6 Conclusions

There is a long standing need for higher quality performance in NLP systems. It is possible that semantic resources richer than WordNet will enable them to break the current quality ceilings. Both statistical and symbolic NLP systems can make use of such semantic knowledge. With the increased size of the Web, more and more training data is becoming available, and as Banko and Brill (2001) showed, even rather simple learning algorithms can perform well when given enough data.

In this light, we see an interesting need to develop fast, robust, and scalable methods to mine semantic information from the Web. This paper compares and contrasts two methods for extracting *is-a* relations from corpora. We presented a novel pattern-based algorithm, scalable to the terascale, which outperforms its more informed syntactical co-occurrence counterpart on very small and very large data.

Albeit possible to successfully apply linguistically-light but data-rich approaches to some NLP applications, merely reporting these results often fails to yield insights into the underlying theories of language at play. Our biggest challenge as we venture to the terascale is to use our new found wealth not only to build better systems, but to improve our understanding of language.

Table 10. QA definitional evaluations for co-occurrence-based system.

	TOP-1		TOP5	
	Strict	*Lenient*	*Strict*	*Lenient*
1.5MB	0%	0%	0%	0%
15MB	0%	0%	0%	0%
150MB	0%	0%	0%	0%
1.5GB	6.0%	8.0%	6.0%	8.0%
6GB	36.0%	44.0%	60.0%	62.0%

References

Banko, M. and Brill, E. 2001. Mitigating the paucity of data problem. In Proceedings of HLT-2001. San Diego, CA.

Berland, M. and E. Charniak, 1999. Finding parts in very large corpora. In *ACL-1999*. pp. 57–64. College Park, MD.

Brill, E., 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–566.

Brill, E.; Lin, J.; Banko, M.; Dumais, S.; and Ng, A. 2001. Data-intensive question answering. In *Proceedings of the TREC-10 Conference*, pp 183–189. Gaithersburg, MD.

Caraballo, S. 1999. Automatic acquisition of a hypernym-labeled noun hierarchy from text. In *Proceedings of ACL-99*. pp 120–126, Baltimore, MD.

Curran, J. and Moens, M. 2002. Scaling context space. In Proceedings of ACL-02. pp 231–238, Philadelphia, PA.

Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 191 (1993), 61–74.

Etzioni, O.; Cafarella, M.; Downey, D.; Kok, S.; Popescu, A.M.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 2004. Web-scale information extraction in Know-It All (Preliminary Results). To appear in the *Conference on WWW*.

Fleischman, M.; Hovy, E.; and Echihabi, A. 2003. Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of ACL-03*. pp. 1–7. Sapporo, Japan.

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*. pp. 80–87. Edmonton, Canada.

Harris, Z. 1985. Distributional structure. In: Katz, J. J. (ed.) *The Philosophy of Linguistics*. New York: Oxford University Press. pp. 26–47.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In COLING-92. pp. 539–545. Nantes, France.

Hindle, D. 1990. Noun classification from predicate-argument structures. In *Proceedings of ACL-90*. pp. 268–275. Pittsburgh, PA.

Lin, D. 1994. Principar - an efficient, broad-coverage, principle-based parser. *Proceedings of COLING-94*. pp. 42–48. Kyoto, Japan.

Lin, D. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL-98*. pp. 768–774. Montreal, Canada.

Mann, G. S. 2002. Fine-Grained Proper Noun Ontologies for Question Answering. *SemaNet' 02: Building and Using Semantic Networks*, Taipei, Taiwan.

Miller, G. 1990. WordNet: An online lexical database. *International Journal of Lexicography*, 3(4).

Och, F.J. and Ney, H. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL*. pp. 295–302. Philadelphia, PA.

Pantel, P. and Lin, D. 2002. Discovering Word Senses from Text. In *Proceedings of SIGKDD-02*. pp. 613–619. Edmonton, Canada.

Pantel, P. and Ravichandran, D. 2004. Automatically labeling semantic classes. In *Proceedings of HLT/NAACL-04*. pp. 321–328. Boston, MA.

Riloff, E. and Shepherd, J. 1997. A corpus-based approach for building semantic lexicons. In *Proceedings of EMNLP-1997*.

Voorhees, E. 2003. Overview of the question answering track. In *Proceedings of TREC-12 Conference*. NIST, Gaithersburg, MD.