Automatic Extraction of Turkish Hypernym-Hyponym Pairs From Large Corpus

Savaş Yıldırım and Tuğba Yıldız
Istanbul Bilgi University
Department of Computer Engineering
Engineering Faculty
savasy@bilgi.edu.tr, tdalyan@bilgi.edu.tr

ABSTRACT
In this paper, we propose a fully automatic system for acquisition of hypernym/hyponymy relations from large corpus in Turkish Language. The method relies on both lexico-syntactic pattern and semantic similarity. Once the model has extracted the seeds by using patterns, it applies similarity based expansion in order to increase recall. For the expansion, several scoring functions within a bootstrapping algorithm are applied and compared. We show that a model based on a particular lexico-syntactic pattern for Turkish Language can successfully retrieve many hypernym/hyponym relations with high precision. We further demonstrate that the model can statistically expand the hyponym list to go beyond the limitations of lexico-syntactic patterns and get better recall. During the expansion phase, the hypernym/hyponym pairs are automatically and incrementally extracted depending on their statistics by employing various association measures and graph-based scoring. In brief, the fully automatic model mines only a large corpus and produces is-a relations with promising precision and recall. To achieve this goal, several methods and approaches were designed, implemented, compared and evaluated.

KEYWORDS: hypernym/hyponym, lexico-syntactic patterns.

Proceedings of COLING 2012: Demonstration Papers, pages 493–500,
COLING 2012, Mumbai, December 2012.

1 Introduction

In this study, we describe how to acquire hypernym/hyponymy relations from a Turkish corpora (Sak et al., 2008) in a fully automatic way. The system extracts possible hypernym/hyponym pairs by using lexico-syntactic patterns, then it expands the hyponym list depending on semantic similarity.

The Hypernym/Hyponym relation is one of the semantic relations that play an important role for NLP. The terms hyponym and hypernym have the definition summarized as "hyponym is (a kind) of hypernym" (Miller et al., 1990). In recent years, many approaches have been developed to build semantic lexicons and extract the relations from a corpus or a dictionary. Hand-built lexicons, such as Cyc (Lenat et al., 1986) and WordNet (Miller et al., 1990; Miller, 1995; Fellbaum, 1998), are the most useful to provide resources for NLP applications. Some attempts (Markowitz et al., 1986; Alshawi, 1987; Jensen and Binot, 1987; Ahlswede and Evens, 1988) used patterns to extract semantic relation from a dictionary. Hearst was the first to apply a pattern-based method (Hearst, 1992, 1998). Several researchers have also used corpus-driven and pattern-based methods(Rydin, 2002; Cederberg and Widdows, 2003; Ando et al., 2004; Snow et al., 2004; Sang and Hofmann, 2007; Caraballo, 1999; Alfonseca and Manandhar, 2001; Etzioni et al., 2004; Ritter et al., 2009). Pattern-based methods have also been applied to web documents (Pasca, 2004; Kozareva et al., 2008; Kozareva and Hovy, 2010). There have been significant studies which present statistical and graph-based methods (Chodorow et al., 1985; Widdows and Dorow, 2002; Sumida and Torisawa, 2008; Imsombut and Kawtrakul, 2008).

Few studies have been published for Turkish Language, BalkaNet (Bilgin et al., 2004) is the first WordNet project for Balkan languages such as Turkish, although the project has not yet been completed. Some attempts used a Turkish Dictionary, TDK [1]. (Yazıcı and Amasyalı, 2011; Güngör and Güngör, 2007; Orhan et al., 2011; Şerbetçi et al., 2011) All studies of semantic relation are mostly based on a Turkish dictionary. Our study is the major corpus-driven attempt at integrating lexico-syntactic patterns and a bootstrapping approach.

2 The Methodology

Once the system has simply extracted possible hypernyms by using lexico-syntactic patterns from a Turkish corpus of 490M tokens, it incrementally expands the list by using a bootstrapping algorithm. It uses the most reliable pattern to determine the hypernym/hyponym pairs. For each hypernym, the most reliable candidate hyponyms (seeds) are passed to the bootstrapping algorithm. The algorithm incrementally expands the seeds by adding new seeds depending on a scoring function. The approach employs two different patterns; one is a lexico-syntactic pattern to obtain is-a pairs and the second is a syntactic pattern to compute co-occurrence of the words in a fine-grained way.

2.1 Candidate Hypernym/Hyponym

The most important lexico-syntactic patterns for Turkish are:

1. "NPs gibi CLASS" (CLASS such as NPs),
2. "NPs ve diğer CLASS" (NPs and other CLASS)
3. "CLASS lArdAn NPs" (NPs from CLASS)
4. "NPs ve benzeri CLASS" (NPs and similar CLASS)

[1] The Turkish Language Association

The most reliable pattern is the first pattern that matched over 200,000 cases in the corpus from which 500 reliable hypernyms could be compiled.

2.2 Elimination Rules

Some incorrect hyponyms are extracted due to some factors. The objective of this step is to exclude these kinds of non-hyponyms and to acquire more reliable candidates. A partial exclusion can be performed as follows;

- In the first pattern,we observed that real hyponyms tended to appear in the nominative case. The rule implies if a noun was not in nominative case, it would be eliminated.
- If an item occurs only in a single match with the pattern, it will be eliminated. The assumption is that some matches to the pattern are accidental.
- The more general a word is, the more frequent it is. This rule is that, if a candidate hyponym has a higher frequency (df) than its hypernym, it will be ignored.

2.3 Statistical Expansion

Filtered hyponym list can remain some erroneous candidates. To improve precision, we can sort the candidates by their pattern frequency. The first K of these words can then be used as original seeds for expansion phase, where K can be experimentally chosen (e.g. 5). The system expands the seeds recursively by adding new seeds one by one. The algorithm will stop producing when it reaches sufficient number of items.

Bootstrapping Algorithm: The algorithm is designed as shown in FIGURE1-A. It first extracts hypernym/hyponym pairs and then applies bootstrapping with a scoring function, where a-scoring-f denotes an abstract scoring function for selecting new hyponym candidate. Our scoring methodologies can be categorized in two groups. The first is based on a graph model, the other simply uses semantic similarity between candidates and seeds. We call the former *graph-based scoring* and the latter *simple scoring*. All scoring functions take a list of seeds and propose a new seed.

Graph-Based Scoring: Graph-based algorithms define the relations between the words as a graph. Each word is represented as a vertex and the relation between the words is represented as weighted edge. Some researchers proposed a similar approach (Widdows and Dorow, 2002). Graph-based scoring was implemented as in FIGURE1-B in which each neighbor is compared not only with seed words but also with other neighbors to avoid infections. **Simple Scoring:** This method employs only the edge information between each candidate and the seeds. Therefore, the candidate which is the closest to the centroid of all seeds will be the winner. As shown in FIGURE1-C, the algorithm computes the similarity between a candidate and the seeds.

Edge Weighting: Both graph-based and simple scoring functions employ a similarity measurement to make a decision. Edge weighting schema that we used in the study are as follows:

1. **IDF/co-occur**: co-occurrence * inverse document frequency (IDF) of candidate.
2. **Binary**: If a seed and a candidate co-occur at least once in the corpus: 1, else 0.
3. **Dice**: $occur(i,j)/(freq_i + freq_j)$ where $occur(i,j)$ is the number of times the $word_i$ and $word_j$ co-occur together, and $freq$ is the number of times a $word$ occurs in corpus.
4. **Cosine similarity**: To compute the similarity between the words, a word space model in which words are represented as vectors is used.

Bootstrapping Algorithm (A)

```
Definitions:
INPUT: C, P
OUTPUT: hyponym/hypernym pairs

for each h: H
  cand<-empty
  for each hyponym:hyponyms(h)
     if(pass the elimination)
          cand  <- add hyponym;
  seeds <-take first K cand;
  while (insufficient)
      add-new-one(seeds, a-scoring-f);
  store(h, final-seeds);
```

Graph-Based Scoring (B)

```
Definitions:
INPUT: S
OUTPUT: new seed

for each n in N(S)
  for each m in N(S)
     if n != m
        score+= edge(n,m);
   assign(score, n);
 rank the N(S) by score
 return the best in N(S)
```

Simple Scoring (C)

```
Definitions:
INPUT: S
OUTPUT: new seed

for each n in N(S)
 for each seed in seeds
     score+= edge(n,seed);
 assign(score, n);
rank the N(S) by score
return the best in N(S)
```

Figure 1: Bootstrapping Algorithm and Scoring Functions, where C: Corpus, P: Pattern, H: Hypernym List, S: Seeds, N(S): Neighbors of S

Building the Graph and Co-occurrence Matrix: The words can be represented in a matrix. $cell_{ij}$ represents the number of times $word_i$ and $word_j$ co-occur together. The matrix is a simple representation of a graph. Co-occurrence can be measured with respect to sentences, documents, or a given window of any size. The conventional way to compute co-occurrence is to use all neighbors within a window by eliminating stop words. This approach has proved to be good at capturing sense and topical similarity (Manning and Schütze, 1999). For example, *train* and *ticket* can be found to be highly similar by this method. However, we need to apply more fine-grained methodologies to capture words sharing the same type such as train and auto or ticket and voucher.

To obtain such type similarity, the solution is to use syntactic patterns for computation of co-occurrence. For instance, nouns are considered similar when they are in particular patterns such as *"N and N"* or *"N,N,...,N and N"*. A similar approach was also used by (Cederberg and Widdows, 2003). Words (nouns) considered similar would either all be subject, or all object or all indirect object. This approach makes the model more fine-grained than other conventional ways of computing bi-grams.

3 Experimental Setup and Implementation

We implemented a utility program which can be used to verify and reproduce the results presented in the paper. We used a web corpus of 490M tokens and a morphological parser as language resources (Sak et al., 2008). The model parses the corpus and converts each tokens into the form of surface/lemma/pos. For the experiment and evaluation, the most frequently occurring hypernyms is selected. All settings are described as follows:

1. Lexico-syntactic pattern **(pattern)**: After extracting instances, some candidates are eliminated by elimination rules as described before.
2. Graph Scoring/binary **(gr-bin)**: All distance/edges of the graph are weighted in a binary way.
3. Graph Scoring/co-occurence **(gr-co)**: The edges of the graph are weighted by co-occurrence of words.
4. Simple Scoring/binary **(sim-bin)**: Distance between words is 1, if they co-occur; else 0.

5. Simple Scoring/dice **(sim-dice)**: Edges are weighted by dice coefficient.
6. Simple Scoring/co-occurrence**(sim-co)**: Edges are weighted by the co-occurrence frequency between words.
7. Simple Scoring/cosine **(sim-cos)**: The words are represented as vectors in a matrix. Edge is the cosine similarity between word vectors.

4 Results and Evaluation

For the evaluation phase, we checked the model against 17 selected hypernyms; **country, city, mineral, sport, illness, animal, fruit, bank, event, vegetable, newspaper, tool, profession, device, drink, sector and organization**. In order to measure the success rate, we manually extracted all possible hyponyms of all the classes.

Category	# of output	pattern	gr-bin	gr-co	sim-bin	sim-dice	sim-co	sim-cos	avg
bank	13	84	100	100	100	100	100	100	98
mineral	12	91	100	100	91	100	100	100	97
news.	21	90	52	42	57	47	61	61	59
...	...	...	...	...	...				
Average	43	90	76	75	73	75	78	70	77

Table 1: Precision of the first experiment (# of output of the pattern module)

We tested the system within the seven different settings described above. The **pattern** extracted a number of hypernym/hyponym pairs. The expansion algorithms take the first five candidates as initial seeds (IS) suggested by the pattern module, then expands them to the size of the pattern capacity. Looking at TABLE 1, it seems that pattern module outperforms other expansion algorithms in terms of precision. In order to improve recall, we conducted a second experiment; the expansion algorithms expand IS to the size of actual hyponym list rather than the pattern capacity. And we eventually get a better recall value as shown in TABLE 2.

As third experiment, we incrementally altered the number of IS to investigate changes in recall. We used 10, 15, 20, 25, 30 and the pattern capacity as IS size. The pattern capacity is the number of the entire output proposed by the pattern. The average results are shown in TABLE 3. The results indicate that increasing IS gets better accuracy. This is because pattern module indeed gives promising results but it is limited. TABLE 1 shows that the average score of the pattern is % 90. Since this accuracy is reliable, the expansion algorithms can simply and reliably exploit the outputs of the pattern algorithm.

Category	# of output	pattern	gr-bin	gr-co	sim-bin	sim-dice	sim-co	sim-cos	avg
country	153	86	84	87	85	67	84	80	82
city	88	95	81	88	38	96	77	97	82
...	...	...	...	...	...	...	...		
news.	32	59	46	28	50	34	50	53	46
Average	71	49	59	57	58	56	62	52	56

Table 2: Recall value of second experiment (# of output is the size of actual hyponym list)

There is no significant differences between the accuracy of the different expansion algorithms. **gr-bin**, **sim-bin** and **sim-co** seem to be the best scoring functions. The graph-based algorithms and cosine similarity weighting are costly and time-consuming. We computed the bi-gram

information and weighted our graph by using specific syntactic pattern in a more fine-grained manner. It means only the words co-occurring in "N,N and N" pattern are accepted as bigram. Therefore *sim-co* or *sim-bin* which simply computes the relation are very successful. When looking a troublesome hypernyms having low accuracy in all tables, we face a classical word sense problem. Depending on the sense distribution, the expansion algorithm changes the direction of sense into frequently used senses.

Conclusion

In this paper, we proposed a fully automatic model based on syntactic patterns and semantic similarity. We utilized two patterns: First, the most productive and reliable lexico-syntactic pattern was used to discover is-a hierarchy. We observed that hypernym/hyponm pairs are easily extracted by means of the pattern for Turkish Language. In order to get more precision, we designed some elimination criteria. It gave higher precision but a limited number of pairs with low recall. Second, syntactic pattern was used to compute co-occurrence and expand the list to get higher recall. To discover more hyponyms, we designed a bootstrapping algorithm which incrementally enlarged the pair list.

# of IS	#out	pattern	gr-bin	gr-co	sim-bin	sim-dice	sim-co	sim-cos	average
5	43	**89,7**	76,2	75,4	73,3	74,8	78,0	69,9	77,0
5	71	48,5	59,2	57,0	58,2	55,9	**62,5**	52,1	52,6
10	71	48,5	62,2	59,1	61,9	57,5	**64,2**	53,5	57,9
15	71	48,5	**64,8**	62,1	**66,5**	58,6	**66,9**	56,2	60,4
20	71	48,5	**66,8**	65,6	**67,4**	61,2	**67,6**	62,3	62,1
25	71	48,5	**67,9**	**66,5**	**68,6**	63,1	**69,3**	63,0	63,1
30	71	48,5	**68,8**	**67,4**	**69,9**	64,2	**70,1**	63,7	63,9
all	71	48,5	**70,6**	69,5	**72,5**	66,5	**71,6**	66,4	65,3

Table 3: Third experiment (IS: Initial Seed, all: all output from pattern)

In this modular system, we conducted several experiments to analyze is-a semantic relation and to find the best setup for the model. When we look at the third experiment as shown in TABLE 3, pattern algorithm gave promising results. This module successfully built initial seeds. In order to solve the recall problem, we improved the model capacity to discover new candidates. Both graph-based and simple scoring methodologies were applied and we observed that both approaches had a good capacity to get higher recall, such as 71.6 and 72.5.

A real application could be designed as follows: the all reliable candidates proposed by the pattern method might be used as initial seeds to make the model more robust. Moreover, the pattern module can be refined to obtain more secure candidates. For the sake of simplicity, a simple scoring method with binary weighting **(sim-bin)** would be the best setup with respect to the results.

The results showed that the fully automated model presented in the paper successfully disclose is-a relations by mining a large corpus. In future work, we will design a preprocessing phase in order to avoid the problems coming from polysemy and other factors.

References

Ahlswede, T. and Evens, M. (1988). Parsing vs. text processing in the analysis of dictionary definitions. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 217–224, Buffalo, New York, USA. Association for Computational Linguistics.

Alfonseca, E. and Manandhar, S. (2001). Improving an ontology refinement method with hyponymy patterns. In *Language Resources and Evaluation (LREC-2002)*, Las Palmas, Spain.

Alshawi, H. (1987). Processing dictionary definitions with phrasal pattern hierarchies. *Computational Linguistics*, 13:13–3.

Ando, M., Sekine, S., and Ishizaki, S. (2004). Automatic extraction of hyponyms from japanese newspapers. using lexico-syntactic patterns. In *LREC*. European Language Resources Association.

Bilgin, O., Çetinoğlu, Ö., and Oflazer, K. (2004). Building a wordnet for turkish. volume 7.

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

Cederberg, S. and Widdows, D. (2003). Using lsa and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *In Proceedings of CoNLL*, pages 111–118.

Chodorow, M. S., Byrd, R. J., and Heidorn, G. E. (1985). Extracting semantic hierarchies from a large on-line dictionary. In *Proceedings of the 23rd Annual Meeting of the ACL*, pages 299–304, Chicago, IL.

Şerbetçi, A., Orhan, Z., and İlknur Pehlivan (2011). Extraction of semantic word relations in turkish from dictionary definitions. In *Proceedings of the ACL 2011 Workshop on Relational Models of Semantics*, pages 11–18, Portland, Oregon, USA. Association for Computational Linguistics.

Etzioni, O., Cafarella, M. J., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2004). Web-scale information extraction in knowitall: (preliminary results). In *WWW*, pages 100–110.

Fellbaum, C., editor (1998). *WordNet: an electronic lexical database*. MIT Press.

Güngör, O. and Güngör, T. (2007). Türkçe bir sözlükteki tanımlardan kavramlar arasındaki üst-kavram İlişkilerinin Çıkarılması. volume 1, pages 1–13.

Hearst, M. (1998). WordNet: An electronic lexical database and some of its applications. In Fellbaum, C., editor, *Automated Discovery of WordNet Relations*. MIT Press.

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

Imsombut, A. and Kawtrakul, A. (2008). Automatic building of an ontology on the basis of text corpora in thai. *Language Resources and Evaluation*, 42(2):137–149.

Jensen, K. and Binot, J.-L. (1987). Disambiguating prepositional phrase attachments by using on-line dictionary definitions. *Comput. Linguist.*, 13(3-4):251–260.

Kozareva, Z. and Hovy, E. H. (2010). A semi-supervised method to learn and construct taxonomies using the web. In *EMNLP'10 in Proceedings of Conference on Empirical Methods in Natural Language Processing*, Boston.

Kozareva, Z., Riloff, E., and Hovy, E. H. (2008). Semantic class learning from the web with hyponym pattern linkage graphs. In *ACL'08: HLT in Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1048–1056, Columbus, USA.

Lenat, D., Prakash, M., and Shepherd, M. (1986). Cyc: Using common sense knowledge to overcome brittleness and knowledge acquistion bottlenecks. *AI Mag.*, 6(4):65–85.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Markowitz, J., Ahlswede, T., and Evens, M. (1986). Semantically significant patterns in dictionary definitions. In *Proc. 24rd Annual Conf. of the ACL*, pages 112–119.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244.

Orhan, Z., İlknur Pehlivan, Uslan, V., and Önder, P. (2011). Automated extraction of semantic word relations in turkish lexicon. *Mathematical and Computational Applications, Association for Scientific Research*, (1):13–22.

Pasca, M. (2004). Acquisition of categorized named entities for web search. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 137–145, New York, NY, USA. ACM.

Ritter, A., Soderl, S., and Etzioni, O. (2009). What is this, anyway: Automatic hypernym discovery. In *In Proceedings of AAAI-09 Spring Symposium on Learning*, pages 88–93.

Rydin, S. (2002). Building a hyponymy lexicon with hierarchical structure. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pages 26–33, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Sak, H., Güngör, T., and Saraçlar, M. (2008). Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *GoTAL 2008*, volume 5221 of *LNCS*, pages 417–427. Springer.

Sang, E. T. K. and Hofmann, K. (2007). Automatic extraction of dutch hypernym-hyponym pairs. In *Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands*.

Snow, R., Jurafsky, D., and Ng, A. Y. (2004). Learning syntactic patterns for automatic hypernym discovery. In *NIPS*.

Sumida, A. and Torisawa, K. (2008). Hacking wikipedia for hyponymy relation acquisition. In *In Proceedings of IJCNLP 2008*, pages 883–888.

Widdows, D. and Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In *In 19th International Conference on Computational Linguistics, COLING*, pages 1093–1099.

Yazıcı, E. and Amasyalı, M. F. (2011). Automatic extraction of semantic relationships using turkish dictionary definitions. volume 1, pages 1–13.