EXTRACTION OF HYPONYMY, MERONYMY, AND ANTONYMY RELATION PAIRS: A BRIEF SURVEY

Gurkan Sahin

Department of Computer Engineering, Yildiz Technical University, Istanbul, Turkey

ABSTRACT

Semantic dictionaries are widely used in natural language processing studies. Broad range of methods have been proposed to construct semantic dictionaries until this time. In this study, a brief summary for semantic relation pair extraction is introduced. An overview of selected approaches is given and results are compared to each other for hyponymy, holonymy, and antonymy relations.

KEYWORDS

Semantic Relation, Hyponymy,Hypernymy, Meronymy, Holonymy, Antonymy, Word2Vec, Information Retrieval

1. INTRODUCTION

Automatic extraction of semantic relations pairs from various sources is one of the popular topics in natural language processing (NLP).In these studies, corpus, web pages, dictionary definitions etc. are used as source. The main purpose of these studies is to create a structural semantic dictionary in which the words have various relations with each other. Semantic dictionaries are used in many NLP studies such as document classification, information retrieval etc. WordNet [1] is the best example of semantic dictionary for English. There are many semantic relationships in WordNet such as hyponymy, meronymy, synonymy, antonymy etc. The words in noun, adjective, verb, and adverb in the WordNet are clustered together with synonyms of each other called synset. There are about 117,000 synsets in WordNet and these synsets are interconnected by various semantic relations.

Although these dictionaries provide great benefits for NLP applications, their creation by people is very time consuming. Various methods have been proposed for automatic creation of semantic dictionaries using computer programs. The most commonly used method is pattern-based approach. Relation pairs can be easily extracted using some semantic relation patterns. In addition to pattern-based method, corpus statistics and machine learning algorithms are also used to determine semantic relations. In this study, some examples from literature studies about hyponymy, meronymy, and antonymy relations are given.

2. HYPONYMY RELATION

Hyponymy represents a semantic relationship between a generic and specific term. The generic term is called hypernym and the specific term is called hyponym. Hyponymy relationship can be represented by "X is a kind of Y" pattern. In this pattern, X and Y represent any hyponym and hypernym term such as apple-fruit, dog-animal, respectively. Hyponymy is an asymmetrical relationship. While "each X is a/an Y" condition is true, the reverse (each Y is a/an X) is not true. Therefore, X and Y cannot replace with each other. Hyponymy is a transitive semantic relation. If

DOI: 10.5121/ijnlc.2017.6201

X is a hyponym of Y, and Y is a hyponym of Z, then X is a hyponym of Z. Given two propositions, “cat is an animal” and “animal is a living creature”, “cat is a living creature” can be extracted from combining of these two propositions. Hyponyms and hypernyms can be represented in a tree structure using the transitivity. In the tree structure, while lower levels represent more specific terms, higher levels represent more general terms. In the hierarchical structure, a hyponym can be a hypernym and a hypernym can be a hyponym at the same time. Given two propositions “apple is a fruit” and “fruit is a food”, while fruit is hypernym of apple, also fruit is hyponym of food. In the hierarchical structure, same level sub-nodes of given a node are called co-hyponyms. For example, cat, dog, bird are hyponyms for “animal” hypernym, are also co-hyponyms of each other.

Hearst [2] performed first study on extracting hyponym-hypernym pairs from Grolier’s American Academic Encyclopedia (Grolier 1990) corpus. Firstly, Hearstmanually determinedhigh frequent hypernymy patterns in corpus, then these patterns were used to generate new hyponym-hypernym pairs. Aim of the study was extracting new pairs which are not exist in human created dictionaries and also enriching the contents of the dictionaries. If we take Hyponym(broken bone, injury) example, while it is very unlikely that the pair is contained in a hand-made dictionary, similar pairs can be easily extracted from the texts using Hearst’spatterns.

Table 1. Hearst’shyponymy patterns [2]

Pattern	Example
NP_0 such as {NP_1, NP_2…, (and | or)} NP_n	-
such NP as {NP,} * {(or | and)} NP	… such authors as Herrick, Goldsmith, and Shakespeare.
NP {, NP} * {,} or other NP	Bruises, wounds, broken bones or other injuries…
NP {, NP} * {,} and other NP	… temples, treasuries, and other important civic buildings.
NP {,} including {NP,} * {or | and} NP	All common-law countries, including Canada and England…
NP {,} especially {NP,} * {or | and} NP	… most European countries, especially France, England, and Spain.

Using the patterns in Table 1, hyponym-hypernym pairs were obtained in Hyponym(N_0, N_1) structure. Here, N_0 represents hyponym and N_1 represents hypernym. To measure success of the method, extracted pairs were compared with WordNet (Miller et al., 1990) pairs and three possible situations were arisen.

Verify: If N_0 and N_1 concepts are exist in WordNet and also Hyponym (N_0, N_1) relation exists in the WordNet name hierarchy, Hyponym(N_0, N_1) relation is verified.

Critique: If N_0 and N_1 concepts are exist in WordNet and Hyponym(N_0, N_1) relation does not exist in the WordNet name hierarchy, it can be suggested to establish new hyponym-hypernym concept relation links in WordNet using Hyponym(N_0, N_1).

Augment: If one or both of the N_0 and N_1 concepts are not in WordNet, N_0-N_1 and relationship between them can be suggested as a new input to WordNet.

Using Grolier’s American Academic Encyclopedia, 152 Hyponym(N_0, N_1) relations were extracted. In addition, 46 relationships were found using “New York Times text”. Although the number of relations obtained was small when compared with the size of texts used, it was said that results are promising.

Rydin [3] used a compilation of 293,692 Swedish daily news articles to create hyponym-hypernym concept dictionary.The corpus used in the study was not limited to a single domain. First, all words in the corpus were tagged (part-of-speech tagged) and lemmatized. Five lexical patterns were identified and these patterns were used to extract noun-noun type hyponym-hypernym concept pairs from different domains corpus. Some of the sentences containing 5 patterns were examined and it was observed that 92% of the pairs accurately reflected hyponymy relation. Thus, the reliability of the patterns was shown. Also, it was thought that obtained wrong pairs were caused by some NLP problems such as word sense disambiguation, wrong part-of-speech tagging etc. 1,000 pairs were selected from generated hierarchical structure, then their accuracy was evaluated by 4 different people as 67.4%-76.6%.

Table 2. Rydin's Swedish hyponymypatterns [3]

Swedish Patterns
sadana NP_h som ((NP_n,)* NP_2 och | eller) NP_1
NP_h (sa) som ((NP_n,)* NP_2 och | eller) NP_1
NP_1 (, NP_n) * och | eller annan NP_h
NP_1 (, NP_n) * och | eller liknande NP_h
NP_1 (, NP_n) * och | eller NP_2, num.expr. NP_h

Ando [4] automatically extracted hyponym-hypernym pairs from a 32 year Japanese newspaper corpus.The study was conducted in 3 steps. Firstly, various hyponym-hypernym pairs were extracted from the 6 year newspaper corpus. Using these initial pairs, 30 hyponymy patterns were extracted but only 7 reliable and frequent patterns were selected. Examples of unreliable 2 patterns are "A wo hajime B" (A as an example of B) and "A wo fukume B" (B including A).
As a result of the experiments, 49%-87% precision was obtained according to different patterns. In addition, when compared with abstract hypernyms, more accurate hyponyms were obtained from concrete hypernyms.

Table 3. Ando'shyponymypatterns [4]

Japanese pattern	English pattern	Frequency	Precision		
			Concrete	Abstract	Total
A nado B	B such as A	10,277	42%	15%	33%
A nado-no B		16,463	53%	22%	43%
A ni-nita B	B which is similar to A	448	76%	38%	67%
A no-youna B	B which look/sound like A	3,160	46%	17%	36%
A to-iu B	B which is called/named A	14,719	50%	23%	40%
A to-yoba-reru B		481	73%	46%	65%
A igai-no B	B other than A	1,151	84%	25%	70%

7 patterns in Table 3 were applied to 32 year news texts (different from the 6 year newspaper news corpus used to obtain the patterns), then candidate hyponyms were extracted for 130 target hypernyms. For example, to extract hyponyms of *yasai* (vegetable), "X nado-no yasai (vegetables such as X)" pattern was applied to corpus and *tomato* was obtained instead of X hyponym. For each of 130 hypernyms, 30 candidate hyponyms extracted from patterns were examined and precisions in the Table 3 were calculated.

In this study, it was shown that higher precision was obtained from general hypernyms (which are in the middle of the Is-A hierarchy) when compared to special hypernyms (which are in the lower part of the Is-A hierarchy). For example, although 80% precision was obtained for the "norimono (vehicle)", "doubutsu (animal)", and "kagu (furniture)" hypernyms which are in the middle of the Is-A hierarchy, less than 50% precision was obtained for "basu (bus)", "inu (dog)", and "isu (chair)" hypernyms which are in the lower part of the hierarchy.

Extracted new hyponyms were compared with hyponyms which are in the associative concept dictionary created by humans. In this way, it is observed that how many new hyponyms are present in the dictionary and how many are not. 55% (442/810) of the hyponyms in dictionary were seen among the hyponyms extracted from the corpus. In addition, 18% (143/786) of correct hyponyms extracted from the corpus were found in the dictionary. As a result, it was shown that popular hyponyms in the relational concept dictionary were also present in the corpus. Also, many hyponyms which were absent in the dictionary were extracted from the 32 year newspaper corpus.

Snow [5] attempted to learn hyponym-hypernym pairs from the six million word text corpus. Snow said that lexical patterns used by Hearst are only a few of all patterns that represent the Is-A relationship. Also, it was proposed that unlike lexico syntactic patterns, there are hidden patterns which are not directly visible between hyponym-hypernym pairs. For this reason, automatically extraction of these patterns (dependency patterns) instead of manually creating patterns was proposed. To extract dependency patterns, firstly, corpus was parsed by MINIPAR tool and all of the dependency patterns for all sentences were extracted. Then, various hyponym-hypernym pairs which were prepared from WordNet were searched in the parsed corpus and hyponymy dependency patterns (dependency paths) were extracted. The extracted patterns were used as features to classify noun-noun type hyponym-hypernym pairs. For classification of pairs, 10-fold cross validation and different classification algorithms were used and the best result was obtained from the logistic regression classifier. The classified pairs were compared with WordNet pairs and better results were obtained than the previous studies using lexical patterns. In addition, according to the Hearst's lexical patterns based classifier, the F measurement score increased by 132% relative.

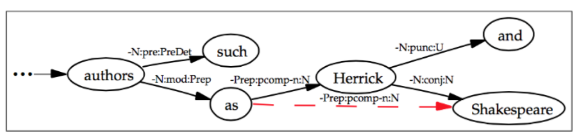

Figure 1. A dependency pattern obtained from MINIPAR [5]

NP_X and other NP_Y:	(*and*,U:PUNC:N),-N:CONJ:N, (*other*,A:MOD:N)
NP_X or other NP_Y:	(*or*,U:PUNC:N),-N:CONJ:N, (*other*,A:MOD:N)
NP_Y such as NP_X:	N:PCOMP-N:PREP,*such_as*,*such_as*,PREP:MOD:N
Such NP_Y as NP_X:	N:PCOMP-N:PREP,*as*,*as*,PREP:MOD:N,(*such*,PREDET:PRE:N)
NP_Y including NP_X:	N:OBJ:V,*include*,*include*,V:I:C,*dummy_node*,*dummy_node*,C:REL:N
NP_Y, especially NP_X:	-N:APPO:N,(*especially*,A:APPO-MOD:N)

Figure 2. Dependency pattern equivalentsof the Hearst's lexico-syntactic patterns [5]

Ritter [6] proposed three different ways to extract hyponym-hypernym concept pairs. In HYPERNYMFINDER$_{freq}$ method, a classification was performed using Hearst'spattern frequencies. 117 million web pages were used in the study and candidate hypernyms which match any of Hearst's patterns were extracted. The accuracy of candidate hypernym is usually related to the pattern matching frequency. However, it was seen that even the hypernyms which co-occur high frequency with patterns can be wrong. The reason for this is that the general structure of the patterns is prone to produce erroneous concepts. Hence, it was said that a simple frequency based thresholding is not sufficient to achieve high precision rate. To test the method, 953 noun type hyponyms were randomly selected from 117 million web pages. For each hyponym, top 5 hypernym proposed from the patterns were manually marked as “right”, “wrong” or “uncertain”. In the $\text{HYPERNYMFINDER}_{SVM}$ method, various features (different pattern frequency, etc.) and SVM classifier were used to classify correct hyponym-hypernym pairs. Because $\text{HYPERNYMFINDER}_{freq}$ and $\text{HYPERNYMFINDER}_{svm}$ methods use Hearst’s patterns, these two methods fail to detect pairs which do not co-occur with patterns. $\text{HYPERNYMFINDER}_{HMM}$ method was proposed to solve this problem. The method used hidden markov model to identify the hypernyms which do not co-occur with the patterns. Using the $\text{HYPERNYMFINDER}_{HMM}$ method, recall value increased from 80% to 82% for noun type hypernyms and from 65% to 71% for private name hypernyms.

Yildiz [7] extracted hyponym-hypernym pairs using 2 different methods from 500 million words Turkish corpus [8]. In the first method, 4 different hypernymy patterns were used to extract noun-noun type pairs. Although reliable patterns were used in the study, extracted pairs may be wrong. For this reason, it is necessary to eliminate the wrong pairs. To eliminate wrong pairs, “the more general a concept is, the higher the corpus frequency” hypothesis was suggested. In other words, it is expected that the corpus frequency of a hypernym concept must be larger than the corpus frequency of its hyponyms. The hypothesis was applied to 1,066 manually generated hyponym-hypernym pairs and 118 pairs did not provide the hypothesis. Thus, it was shown that the elimination method works with nearly 10% error. As a result of the experiments performed, an increase in the precision of pairs was obtained when the elimination method was used. In addition to first elimination, a second elimination method based on statistical similarity was proposed. In this method, semantic similarity between candidate hyponyms and target hypernym was used. Firstly, all neighbor context words of target hypernym were extracted. To find neighbor words, x^2 (chi-square) association score (for 2-gram word associations) was used and 60 neighbors which have the highest x^2 score were selected in total, including 20 “noun”, 20 “adjective”, and 20 “verb” type. Each candidate hyponym and target hypernym were represented by vectors using co-occurrence frequencies together with 60 selected contex words. Then, the cosine similarity between each candidate hyponym vector and target hypernym vector was calculated. Candidates which have greater score than threshold similarity value were classified as correct, while others were eliminated. Four different target hypernyms, namely “fruit”, “vegetables”, “country”, and “fish” were identified and candidate hyponyms were extracted from the patterns. Thanks to the elimination steps, a significant increase in the mean precision value was obtained by 83%. Apart from the elimination based methods, expansion based a second method was proposed to extract hyponym-hypernym pairs by Yildiz [9]. In this method, a small number of hyponym-hypernym pairs were manually created at the beginning,then existing pairs were expanded by adding 1 new pair at each step. When the target number of pairs was reached, the algorithm was terminated.

Table 4. Yildiz’s Turkish hypernymy patterns [7]

Turkish Pattern	Turkish Examples
NPs gibi CLASS	Elma, armut, muz gibi meyveler
NPs ve diğer CLASS	Elma, armut ve diğer meyveler
CLASS1ArdAn NPs	Meyvelerden elma, armut, muz
NPs ve benzeri CLASS	Elma, armut ve benzeri meyveler

Sahin [10] used a pattern-based method to extract hyponym-hypernym pairs from the Turkish corpus. Firstly, various hypernymy patterns were extracted from the corpus and these patterns were used to generate new pairs. Three different scoring methods were used to evaluate the obtained pairs.

Table 5. Sahin's hypernymy patterns (X: hyponym, Y: hypernym) [10]

Turkish pattern	**Corpus frequency of the pattern**
X gibi Y	72,343
X gibi bir Y	8,350
X gibi birçok Y	532
X gibi bazı Y	516
X gibi çeşitli Y	453
X ve diğer Y	8,222
X veya diğer Y	629
X ve benzeri Y	3,528
X ve çeşitli Y	776

Total pattern frequency: It was found that how many times each of extracted pairs co-occurred with all the patterns in corpus. These pairs were sorted according to total pattern frequency in descending order and first k pairs were examined visually.

Different pattern frequency: It was found that how many different patterns co-occurred with each pair. Then, pairs were sorted according to different pattern frequency in descending order and first k pairs were examined visually.

Word2Vec vector similarity: 200-dimensional vector information of each of the corpus words was extracted using word2vec [11]. Cosine vector similarities between candidate hyponyms and target hypernym were calculated. For testing, 15 target hypernyms were identified and candidate hyponyms were extracted. First k sorted hyponyms were examined and labeled according to 3 different methods. 81%(for the total pattern frequency method), 81%(for the different pattern frequency method), and 83%(for the word2vec similarity method) mean precisions were obtained. In this study, word2vec word vector similarities were used for the first time to extract Turkish hyponym-hypernym pairs from corpus and it was experimentally demonstrated that word2vec succeeds in detecting the correct hypernymy pairs. Various Turkish hyponym-hypernym pairs ranked according to word2vec similarity score were given in the Appendix.

Table 6. Comparative summary for hypernymy

Author(Year)	**Method and used resources**	**Result**
Hearst (1992)	- Grolier's American Academic Encyclopedia and New York Times Texts were used as corpus - 6 high frequency hypernymy patterns were selected to extract new pairs - Hearst showed for the first time in this work that by using a small number of manually generated patterns, hyponym-hypernym pairs can be extracted from corpus with high accuracy	- Extracted new pairs are compared with WordNet hypernymy pairs and pairs which are not in the hierarchy were proposed for adding - In this respect, it was aimed to expand existing WordNet (Miller 1990) structure
Rydin (2002)	- A collection of 293,692 Swedish daily news articles from different domains - By using 5 different lexical patterns, a hierarchical structure consisting of hyponym-hypernym pairs was created	- 1,000 pairs in the generated hierarchical structure were selected and 67.4%-76.6% accuracy was obtained
Ando (2004)	- 32 years Japanese newspaper news corpus - 7 hypernymy patterns were used	- 49%-87% precision for 130 target hypernym
Snow (2005)	- 6 million words corpus - It was suggested that Hearst's patterns are insufficient to find most hyponym-hypernym pairs - It was benefited from dependency patterns	- The best classification success was obtained from logistic regression algorithm using 10-fold cross validation - Compared to the classifier

	to solve this problem - All dependency paths were extracted by parsing corpus with MINIPAR - Hypernymy dependency paths were extracted using noun-noun type initial pairs and these patterns were used as feature to classify pairs	created using Hearst's patterns, 132% relative success was achieved in the F measurement score
Ritter (2009)	- 117 million web pages - $\text{HYPERNYMFINDER}_{freq}$ (pairs were evaluated according to a certain pattern frequency threshold value; noisy pairs can come from patterns), $\text{HYPERNYMFINDER}_{SVM}$ (Hearst'spattern frequencies were used to classify pairs), $\text{HYPERNYMFINDER}_{HMM}$ (for pairs which do not co-occur with Hearst'spatterns) methods were used	- Using $\text{HYPERNYMFINDER}_{HMM}$, recall values increased from 80% to 82% for "noun" type pairs and 65% to 71% for "proper noun" type pairs
Yildiz (2012)	- 500 million words Turkish corpus - 4 lexico-syntactic patterns were used - 2 different methods (corpus frequency based and context word similarity based eliminations) were used for elimination of wrong pairs	- A significant increase in the precision value was achieved thanks to the elimination steps - An average of 83% precision was achieved for 4 different target hypernym concepts
Sahin (2016)	- 500 million words Turkish news corpus - 9 different lexico-syntactic patterns were used - 3 different methods (total pattern frequency, different pattern frequency, word2vec vector similarity) were used to evaluate correctness of extracted new pairs	- 81%-83% average precision was obtained for 15 target hypernym concept - The study showed that wor2vec vector similarity score was successful in determining the correct hyponym-hypernym pairs

3. HOLONYMY RELATION

Holonymy represents semantic relationship between a whole term and a part term. In this relation, part of a whole is called meronym and whole of a part is called holonym. Holonymy relationship can be represented by "X is part of Y" and "X is member of Y" patterns. In these patterns, X and Y represent any meronym and holonym term such as wheel-car, leaf-tree etc., respectively. As in hyponymy, holonymy is asymmetric and transitive semantic relation. If X is a meronym of Y and Y is a meronym of Z, then X is a meronym of Z. Given two propositions "nail is part of finger" and "finger is part of arm", "nail is part of arm" can be extracted using transitivity.

Sahin [10], [12] extracted meronym-holonym pairs from Turkish corpus using 5 different patterns.

Table 7. Sahin's holonymy patterns (X:holonym, Y:meronym) [10]

#	Turkish Pattern	Corpus frequency of the pattern
1	X in Y si	549,516
2	X Y si	11,841,159
3	Y si olan X	44,369
4	Y li X	1,234,127
5	Y siz X	170,163

In order to measure success of the system, 18 target holonym concepts were identified and these holonyms were searched together with the patterns to extract candidate meronym concepts.

Similar to [10], total pattern frequency, different pattern frequency and word2vec vector similarity were used in evaluating correctness of candidate meronyms. First 10, 20, 30, 40, and 50 candidate meronyms were ranked according to each scoring method and candidates were visually evaluated. The highest average precision was obtained from the different pattern frequency method with 63%-86%. Unlike antonym pairs, word2vec vector similarity was successful in determining correct meronym-holonym pairs. Various Turkish meronym-holonym pairs ranked according to word2vec similarity score were given in the Appendix.

Yildiz [13] proposed a semi-automatic method for extracting meronym-holonym pairs from Turkish corpus. The methodtakes some initial pairs as input and uses it to extract holonymy patterns. Different association measurement methods (pmi, dice, t-score) were used to determine reliability of patterns and extracted new pairs. Firstly, 200 meronym-holonym pairs were prepared. While 50 out of 200 were used to extract holonymy patterns, remaining 150 pairs were used to determine reliability of the patterns. For all prepared target holonyms, candidate meronyms were extracted using the patterns. All candidates were ranked according to the 3 different reliability score and first 10, 20, and 30 candidates were examined.

Table 8. Yildiz's most reliable regular expression patterns for holonymy [13]

<table>
<tr><th>Pattern name</th><th>Regular expressionof Turkish pattern</th><th>Turkish example</th></tr>
<tr><td>P1</td><td>\w+\+noun[\w\+]+gen +\w+\+noun[\w\+]+p3sg</td><td>arabanın kapısı</td></tr>
<tr><td>P2</td><td>\w+\+noun\+a3sg\+pnon\+nom
\w+\+noun\+[\w\+]+p3sg</td><td>Araba kapısı</td></tr>
<tr><td>P3</td><td>Evin+ev+noun+a3sg+pnon+gen
arka+arka+noun+a3sg+pnon+nom
bahçe+bahçe+noun+a3sg+pnon+nom
kapısı+kapı+noun+a3sg+p3sg+nom</td><td>evin arka bahçe kapısı</td></tr>
<tr><td>P4</td><td>\w+\+noun\+a3pl\+pnon\+abl
birinin\+biri\+pron\+quant\+a3sg\+p3sg\+gen
\w+\+noun\+\w+\+p3sg</td><td>evlerden birinin kapısı</td></tr>
<tr><td>P5</td><td>\w+\+noun[\w\+]+p3sg\+\w+
(\w+\+noun\+a3sg\+pnon\+nom|
\w+\+noun\+a3sg\+pnon\+nom\-adj*with)
(\w+\+verb\+pos\-adj*prespart|
\w+\+verb\+pos\+narr\+a3sg) \w+\+noun\+a3sg</td><td>kapısı kilitli olan ev</td></tr>
<tr><td>P6</td><td>\w+\+noun\+a3sg\+pnon\+nom\-adj*with
\w+\+noun\+</td><td>bahçeli ve havuzlu ev</td></tr>
</table>

Reliability scores were calculated for each pattern using 150 out of 200 initial seeds.

Table 9. Yildiz's pattern reliabilities [13]

	Rel(P1)	Rel(P2)	Rel(P3)	Rel(P4)	Rel(P5)	Rel(P6)
Pmi	1.58	1.53	0.45	0.04	0.07	0.57
Dice	0.01	0.003	0.01	0.004	0.001	0.003
t-score	0.11	0.12	0.022	0.0004	0.001	0.03

Pmi, dice, and t-score scores were used to calculate reliability of the patterns. Five target holonym words were identified and candidate meronyms were evaluated after being scored. The highest average precison values were 72% for the first 10 candidates, 67% for the first 20 candidates, and 64% for the first 30 candidates, respectively.

In [14] Espresso algorithm was proposed for extracting various semantic relation pairs. The Espresso first takes very small number of pairs as input and it learns relation patterns to generate more pairs. It performs these operations in 3 stages iteratively.

1. Extracting of patterns: At the beginning, a small number of meronym-holonym pairs were given to the system and patterns were extracted from corpus.

2. Determining reliability of patterns: While some of extracted patterns may be correct, others may not exactly reflect the relationship. For this reason, it is necessary to select the right relation patterns. Threshold frequency based elimination is a simple method, but not very reliable. When this elimination is used, patterns that are correct but low frequent ones can be eliminated. Instead of raw corpus frequency, pointwise mutual information (PMI) [15] association score between pattern and initial pairs was used to deterimine reliability of the pattern.

(1) was used to calculate reliability of patterns. In (1), $r_\pi(p)$ is reliability score of pattern p and I is number of initial pairs used to produce patters. Max_{pmi} is the maximum pmi value between all patterns and all pairs in the corpus. As i={x,y}, r(i) is reliability score of pair i and pmi(i,p) is pmi score between pattern p and pair i. Since initial pairs were manually generated, all initially r(i) values were equal to 1.

$$r_\pi(p) = \frac{\sum_{i \in I}\left(\frac{pmi(i,p)}{max_{pmi}} * r(i)\right)}{|I|} \qquad (1)$$

(2) was used to calculate pmi(i,p). In (2), |x,p,y| is co-occurrence frequency of (x,y) pair together with pattern p. |x,*,y| is co-occurrencefrequency of (x,y) pair together with all patterns in corpus and |*,p,*| is co-occurrencefrequency of pattern p together with all pairs in corpus. The result is multiplied by -1 because (2) will produce negative value.

$$pmi(i,p) = \log \frac{|x,p,y|}{|x,*,y||*,p,*|} \qquad (2)$$

The Espresso ranked all patterns by their reliability scores and patterns that were lower than threshold reliability score were deleted. The remaining patterns were used to generate new pairs.
3. Determining reliability of new pairs: For new extracted meronym-holonym pairs obtained from reliable patterns, similar reliability score was calculated using (3). In (3), r(i) is reliability score of pair i. |P| is number of reliable patterns found in the previous step and $r_\pi(p)$ is reliability score of pattern p.

$$r(i) = \frac{\sum_{p \in P}\left(\frac{pmi(i,p)}{max_{pmi}} * r_\pi(p)\right)}{|P|} \qquad (3)$$

As a result, meronym-holonym pairs were extracted from TREC-9 [16] corpusconsisting of newspaper news by 80% precision.

4. Antonymy Relation

Antonymy represents opposite semantic relation between a word and the other word or among words in the same part of speech, such as tall-short (adjective-adjective), quickly-slowly (adverb-adverb). In antonymy, words that are opposite of each other are called antonym. Therelationship can be represented by "neither X nor Y" pattern. In this pattern, X and Y represent any antonym pair such as good-bad, big-small, long-short etc. Unlike hyponymy and holonymy, antonymy is symmetrical relationship. X and Y terms can be replaced with each other in the pattern like "neither big nor small" or "neither small nor big".

Lin [17] proposed "patterns of incompatibility" method to distinguish synonyms from other distributionally similar words. In this study, "from X to Y" and "either X or Y" patterns were used and if any X and Y words were seen in at least one of these patterns, it was suggested that these two words are semantically opposite of each other. To prove this hypothesis, co-occurrence frequencies of synonym and antonym pairs with patterns on the web were examined using the AltaVista search engine. Similarity scores were calculated for all prepared synonym and antonym pairs using (4) and pairs with a score greater than 2,000 were classified as synonyms, while others were classified as antonyms. For testing, 80 synonym and 80 antonym pairs were determined from the dictionary. Finally, 86.4% precision and 95.0% recall value were obtained.

$$\mathrm{score}(X, Y) = \frac{\mathrm{hits}(X, \mathrm{NEAR}, Y)}{\sum_{\mathrm{pat} \in P} \mathrm{hits}\big(\mathrm{pat}(X, Y)\big) + e} \tag{4}$$

In (4), hits(X, NEAR, Y) is the frequency of (X, Y) pair at a certain word distance from each other. In (4) P represents two patterns given before, hits(pat(X,Y)) represents co-occurrence frequency of the (X, Y) pair with the patterns and e represents a small (e = 0.0001) constant used to make the divider nonzero.

Turney [18] classified synonyms, antonyms, and semantically similar words by supervised machine learning methods. Previously obtained patterns were used as feature vectors and co-occurrence frequency of pairs with these patterns in 5×10^{10} word corpus was used as feature values as well as. In order to test the system, synonym and antonym pairs were prepared from ESL (English as a second language) questions. Using 10 fold cross validation, 75% accuracy was obtained versus 65.4% majority class accuracy. k * N patterns were used as features to classify synonym and antonym words. There, N represents number of pairs to be classified and k is selected as 20. For example, 20x136 = 2720 patterns were used to classify 136 ESL pairs. As a result of the experiments done, it was seen that selection of different k values did not cause any significant change in the obtained result.

As is known, it is difficult to distinguish synonym and antonym pairs from each other using classical distributional model. Even though antonym words are semantically opposite to each other, surprisingly the neighboring context words are very similar to each other as in the case of synonyms.

Lobanova [19] extracted adjective-adjective type antonym pairs from Dutch corpus. The steps of the process are as follows.

Step 1: Firstly, adjective-adjective type initial pairs (seeds) were prepared.
Step 2: The seeds were searched in corpus and antonymy patterns were extracted.
Step 3: For each antonym pattern, a reliability score was calculated using co-occurrence frequency of patterns with the seeds. Patterns whose reliability score is lower than 0.02 assumed as noisy pattern and these patterns were deleted.
Step 4: Using remaining reliable patterns, new antonym pairs were extracted from the corpus and for each new pair, a reliability score was calculated using (5).

$$A_i = 1 - \prod_j (1 - S_j)^{C_{ij}} \quad (5)$$

In (5), A_i is antonym score of pair i, j is number of reliable antonym pattern, S_j is reliability score of pattern j and, C_{ij} is co-occurrence frequency of pattern j with pair i. The algorithm was run through six iterations and only pairs with a antonym score > = 0.9 were used as initial seeds again in the next iteration. At the end of the sixth iteration, accuracy of the extracted pairs was determined by five human using majority voting as “antonym”, “co-hyponym”, “synonym” or “none”. Contrary to expectations, it was seen that most of the new pairs obtained from the patterns were noun-noun type rather than adjective-adjective. In addition, the extracted new pairs were compared with Dutch WordNet and Dutch dictionaries. It was seen that majority of the new pairs were not found in these sources. This demonstrated success of the work being done to produce new antonyms.

Table 10. The precision values obtained for different reliability scores [19]

	Reliability score of pairs (number of pair)							
Relation	**>=0.6**		**>=0.7**		**>=0.8**		**>=0.9**	
Antonym	28.3%	(54)	27.2%	(49)	35.1%	(26)	45.4%	(20)
Co-hyponym	39.2%	(75)	39%	(70)	43.2%	(32)	38.7%	(17)
Synonym	0.5%	(1)	0.5%	(1)	1.3%	(1)	2.3%	(1)
None	32%	(61)	33.3%	(60)	20.3%	(15)	13.6%	(6)
Total		191		180		74		44

Scheible [20] shown that German adjective-adjective synonym and antonym pairs can be distinguished from each other using distinctive neighboring words and word space model. As a result of the studies done, against 50% baseline accuracy, 70.6% accuracy was obtained in classifying synonym and antonym pairs. Two different hypotheses were proposed.

Hypothesis A: Context words of adjective-adjective type synonyms and antonyms are not similar.
Hypothesis B: Not all just a few specific neighbors are useful to distinguish synonyms from antonyms.

In Hypothesis B, successes of distinguishing synonyms from antonyms using 4 different type context words, which were adjective, noun, adverb, and verb, were examined. It was shown that words in the “noun” type are not very distinctive, because they can co-occur with both synonym and antonym words.

T: unhappy *man, woman, child*, ...
SYN: sad *man, woman, child*, ...
ANT: happy *man, woman, child*, ...

As you can see from the example above, the words {man woman, child, ...} co-occurred both “sad” which is synonym of “unhappy” and “happy” which is antonym of “unhappy”. On the other hand, it was said that “verb” type context words are more successful than “nouns” to distinguish synonyms from antonyms.

T: unhappy man*cry, moan, lament*, ...
SYN: sad man*cry, frown, moan*, ...
ANT: happy man*smile, laugh, sing*, ...

To test the hypothesis, 97 adjective-adjective type synonym and antonym pairs were prepared. “noun” (NN), “adjective” (ADJ), “verb” (VV), and “adverb” (ADV) type neighboring context words which co-occur with pairs were used separately and together (COMB) as feature and obtained classification successes were examined. In order to find co-occurrence frequency of

pairs with neighboring words, 880 million word German web corpus was used. Also, LMI (local mutual information) value was used as vector property values instead of raw co-occurrence frequency. Various algorithms in Weka for classification were used with 10-fold cross validation and despite 50% baseline accuracy, 70.6% accuracy was obtained by using "verb" type context words as feature. When the results were examined, it was experimentally shown that antonym and synonym pairs can be distinguished from each other by using beneficial context words.
Santus [21] tried to differentiate pairs of synonyms and antonyms using the APAnt (average-precision-based measure) method. In antonymy relationship, although antonym words are opposite to each other in terms of meaning, it is clearly seen that they share common context words. Cruse [22] explained this situation with the concept of "paradox of simultaneous similarity and difference".

It is difficult to distinguish synonyms from antonyms using classical methods, because both antonym and synonym pairs often co-occur with similar context words. Contrary to the classical view, Santus suggested that synonym and antonym words can be distinguished by using the most common neighbor context words. To confirm the hypothesis, two target words and most associated context words of the targets were identified. Local mutual information (Evert, 2005) [23] value was used to identify neighboring words and 100 context words with the highest LMI value were selected. Santus suggested that synonyms have more common neighbor words than the antonyms. Santus calculated an APAnt score (6) for the given pair by using the number of joint neighbors and order information of the neighboring words. It was suggested that the higher the APAnt score is, the greater degree of antonymy character of the pair.

$$\mathrm{APAnt} = 1 / \sum_{f \in F_1 \cap F_2} \frac{1}{\min(\mathrm{rank}_1(f_1), \mathrm{rank}_2(f_2))} \tag{6}$$

To test the method, 2,232 pairs consisting of 1,070 antonym and 1,162 synonym pairs were generated. APAnt score was calculated for all pairs and boxplot distributions were examined. The success of the proposed method was compared with baseline method which uses co-occurrence frequency of the pairs.

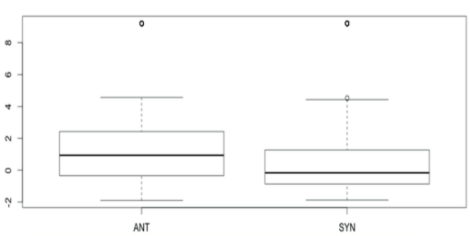

Figure 3. Logarithmic distribution of APAnt values of synonym and antonym pairs [21]

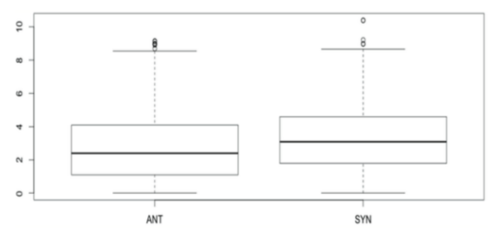

Figure 4. Logarithmic distribution of baseline co-occurrence scores of synonym and antonym pairs [21]

When the boxplot distributions were examined, it was seen that antonym pairs have higher APAnt score distribution than synonym pairs.

Table 11. Average precisions obtained both APAnt and baseline methods [21]

	Antonym	Synonym
APAnt	0.73	0.55
Baseline	0.56	0.74

In this study, it was seen that APAnt method is more successful than baseline method to determine correct antonym pairs.

.

Alyahya et al. [24] extracted noun-noun type Arabic antonym pairs from corpus. Firstly, 57 antonym pairs were prepared and 10 out of 57 with highest frequency were selected to produce antonym patterns. Patterns which co-occur with at least 3 initial pairs were selected and the others were eliminated assuming they were not reliable. LogDice score was examined for each of 10 pairs and scores were found to be greater than 7.

Table 12. Alyahya's antonymy patterns [24]

English equivalents of Arabic patterns	Corpus frequency	How many seeds co-occurred with the pattern
From X to Y	21,040	7
From X or Y	3,351	3
Between X and Y	2,720	5
In X or Y	1,534	3
Neither X nor Y	480	3

Using the patterns, new antonym pairs were extracted from the corpus. Pairs whose LogDice score>= 7 were classified as antonym. 25 extracted pairs were evaluated visually and 76% precision was obtained. While 19 out of 25 were classified as "antonym", 6 out of 25 were classified as "co-hyponym".

Sahin [10] extracted antonym pairs from Turkish corpus using some patterns.

Table 13. Sahin's antonymy patterns [10]

Pattern group	Turkish pattern	Corpus frequency of the pattern group
G1	-X ve Y arasında -X ve Y arasındaki	1,396
G2	-ne X ne Y -ne X nede Y -ne X ne de Y	2,370
G3	-X yada Y -X ya da Y	35,232
G4	-X'den Y'ye	79,363
G5	-X mı Y mı -X mi Y mi -X mu Y mu -X mü Y mü	879
G6	-bir X bir Y	4,251

Firstly, antonym patterns were extracted from corpus using various initial pairs. 80 target words were prepared and these words were searched with patterns to extract candidate antonym of the target. Total pattern frequency (method 1), different pattern frequency (method 2) and word2vec vector similarity (method 3) were used in evaluating the correctness of candidates. All candidates were sorted by descending order according to the 3 evaluation methods and only first candidate was examined visually. As a result of the experiments performed, 79% (for method 1), 0.85% (for method 2) and 59% (for method 3) precisions were obtained. Contrary to expectations, it was experimentally shown that word2vec vector similarity was not very successful in determining the correct antonym pairs.

Table 14. Comparative summary for antonymy

Author (Year)	Method and used resources	Result
Lin (2003)	- Web pages and AltaVista search engine - Patterns of incompatibility, "from X to Y" and "either X or Y" - Pattern frequencies were used to classify pairs	- 86.4% precision and 95% recall were obtained for 80 synonym and 80 antonym pairs
Turney (2008)	- $5x10^{10}$ word corpus - Synonym and antonym pairs were classified by machine learning methods - Various patterns were used as feature and pair-pattern co-occurrence frequency was used as feature	- 136 synonym and 136 antonym pairs were classified by 75% accuracy
Lobanova (2010)	- Dutch corpus - Patterns were extracted using manually created adjective-adjective pairs - For each pattern, a reliability score was calculated and reliable patterns were used to produce new pairs - Contrary to expectations, it was seen that most of the new extracted pairs were in noun-noun type rather than adjective-adjective type - For each new pair, antonymy score was calculated using co-occurrence frequency of pair with all patterns	- 28.3% precision for >=0.6 antonym score and 45.4% precision for >=0.9 antonym score were obtained
Scheible (2013)	- 880 million words German web corpus - The proposed hypothesis assumed that adjective-adjective type German pairs can be distinguished from each other using useful context words - Instead of many context words, fewer and distinctive context words should be used to distinguish synonyms from antonyms - Noun, adjective, adverb, and verb type context words were used - It was shown that "verb" type context words are more distinctive for classification	- 97 synonym and 97 antonym pair were classified using "verb" context words - 70.6% accuracy was obtained versus 50% baseline accuracy
Santus (2014)	- APAnt score was used to classify synonym and antonym pairs - The proposed hypothesis assumed that synonyms co-occur with more common context words than antonyms - The LMI score was used to select context words - High APAnt value represents high degree of antonymy	- 1,070 antonym and 1,162 synonym pairs were used - 73% precision for antonyms and 55% precision for synonyms were obtained from APAnt
Alyahya (2015)	- Noun-noun type antonymy pairs were extracted from Arabic corpus - 20 most frequent pairs in corpus were used to generate reliable antonymy patterns - 5 patterns which co-occur with at least 3 different initial pairs were selected to extract new pairs - Extracted new pairs whose LogDice score greater than 7 were classified as antonym	- 25 new pairs were evaluated visually and 76% precision was obtained
Sahin (2016)	- 500 million word Turkish corpus - 6 different pattern group were used to extract antonym pairs from corpus - Total pattern frequency, different pattern frequency, and word2vec vector similarity score were used to evaluate new pairs	- Maximum 85% precision was obtained for 80 target words - 59% precision was obtained from word2vec vector similarity score

		- Word2vec vector similarity was not found to be very successful to detect correct antonym pairs

5. CONCLUSIONS

This study has summarized some literature studies about hyponymy, holonymy, and antonymy relation pair extraction from various resources.The methods used, the results obtained, the difficulties encountered, and the contributions of the studies have been given comparatively.

ACKNOWLEDGEMENTS

I thank so much my mother Nurhan who has always supported me throughout my life.

APPENDIX

Table 15. Various target hypernyms, candidate hyponyms, word2vec scores, class

Target hypernym/candidate hyponym/word2vec score/class(+ : correct hyponym, - : wrong hyponym)						
science	**fruit**	**sport**	**plant**	**animal**	**vegetables**	**drink**
biology/0.46/+	cherry/0.49/+	riding/0.48/+	vicia sativa /0.54/+	dog/0.61/+	tomato/0.56/+	beverages /0.61/+
zoology/0.43/+	peach/0.44/+	athleticism/0.45/+	thyme/0.53/+	chicken/0.58/+	spinach/0.44/+	beer/0.47/+
anthropology/0.36/+	apple/0.44/+	football/0.44/+	bamboo/0.50/+	cattle/0.56/+	strawberry/0.42/-	lemonade /0.45/+
genetic/0.36/+	rosehip/0.43/+	athlete/0.44/-	moss/0.49/-	sheep/0.49/+	potato/0.41/+	soda/0.43/+
geodesy/0.35/+	grape/0.43/+	gymnastics/0.43/+	blackberry/0.49/+	antelope/0.49/+	broccoli/0.39/+	alcohol/0.43/+
astronomy/0.33/+	tomato/0.43/-	volleyball/0.41/+	mushroom/0.48/-	pig/0.47/+	onion/0.39/+	coffee/0.43/+
sociology/0.33/+	strawberry/0.42/+	fencing/0.39/+	fennel/0.48/+	mammal/0.47/+	salad/0.39/+	brandy/0.39/+
toxicology/0.33/+	apricot/0.40/+	basketball/0.38/+	tree/0.48/+	bird/0.47/+	lettuce/0.38/+	wine/0.38/+
mathematics/0.31/+	pineapple/0.40/+	tae-kwon-do /0.36/+	liquidambar orientalis/0.48/+	cat/0.47/+	beans/0.37/+	buttermilk/0.38/+
epigraphy/0.29/+	avocado/0.39/+	tennis/0.36/+	rosehip/0.47/+	rhino/0.47/+	grape/0.36/-	whiskey/0.37/+
medicine/0.29/+	cherry/0.39/+	karate/0.36/+	hibiscus/0.47/+	cow/0.46/+	cucumber/0.35/+	vodka/0.36/+
paleontology/0.28/+	kiwi/0.37/+	cricket/0.35/+	lavender/0.46/+	lizard/0.45/+	turnip/0.34/+	coke/0.36/+
seismology/0.28/+	mandarin/0.37/+	golf/0.35/+	colutea davisiana /0.45/+	fish/0.45/+	cauliflower/0.33/+	intoxicated/0.36/-
literature/0.28/+	lemon/0.37/-	jogging/0.35/+	oleander/0.43/+	mammoth/0.43/+	roquette/0.33/+	soda water/0.35+

engineering/0.27/+	grapefruit/0.36/+	boxing/0.34/+	ginger/0.39/+	human/0.42/-	leek/0.32/+	fanta/0.34/+
physics/0.27/+	banana/0.36/+	pools/0.34/-	citrus trees/0.39/+	monkey/0.41/+	eggplant/0.31/+	raki/0.33/+
sociologist/0.26/-	hazelnut/0.35/-	surfing/0.34/+	cactus/0.39/+	sea dog/0.41/+	celery/0.31/+	tea/0.30/+
politics/0.25/+	melon/0.35/+	shooting/0.34/+	basil/0.38/+	turkey/0.41/+	mushroom/0.31/+	rosehip/0.29/+
geology/0.25/+	blackberry/0.35/+	aerobic/0.33/+	grape/0.38/+	calf/0.41/+	cabbage/0.30/+	sahlep/0.29/+
chemistry/0.25/+	mulberry/0.33/+	barbell/0.31/+	grain/0.36/+	mouse/0.41/+	pea/0.30/+	compote/0.26/+

Table 16. Various target holonyms, candidate meronyms, word2vec scores, class

Target holonym/candidate meronym/word2vec score/class(+ : correct meronym, - : wrong meronym)						
car	**computer**	**ship**	**hospital**	**school**	**plant**	**bus**
wheel/0.41/+	software/0.66/+	boat/0.69/+	morgue/0.55/+	student/0.59/+	harvest/0.48/+	midibus/0.61/-
garage/0.40/+	desktop/0.51/-	port/0.61/+	clinic/0.47/+	high school /0.56/+	family/0.46/+	passenger/0.54/+
engine/0.40/+	keyboard/0.51/+	deck/0.55/+	policlinic/0.44/+	parent/0.52/+	pathogen/0.44/+	station/0.41/+
service/0.39/+	database/0.49/+	crew/0.54/+	medicine/0.43/+	classroom/0.51/+	nectar/0.43/+	driver/0.38/+
license tag/0.37/+	modem/0.46/+	wharf/0.49/+	treatment/0.40/+	dorm/0.49/+	seed/0.42/+	route/0.38/+
steering wheel /0.37/+	processor/0.45/+	chief engineer /0.48/+	patient/0.40/+	education/0.45/+	fertilizer/0.40/+	convoy/0.36/-
horn/0.36/+	palm/0.45/-	bow/0.48/+	head nurse/0.37/+	teacher/0.43/+	food/0.40/+	ticket/0.35/+
convoy/0.36/-	floppy disk/0.43/+	shipyard/0.47/+	operating room /0.37/+	child/0.39/+	pigment/0.39/+	garage/0.33/+
exhaust/0.36/+	synchronization /0.43/+	tonnage/0.46/+	service/0.37/+	class/0.38/+	fruit/0.39/+	bus terminal /0.32/-
driving license /0.35/+	monitor/0.43/+	fleet/0.46/+	doctor/0.35/+	canteen/0.37/+	fossil/0.39/-	trailer/0.31/+
runway/0.34/-	device/0.41/-	national flag /0.45/+	chief physician /0.34/+	girl/0.36/+	flavor/0.38/+	station/0.31/+
luggage/0.34/+	computer worm /0.39/+	asbestos/0.44/-	prison/0.31/-	curriculum/0.35/+	species/0.38/+	bus assistant /0.31/+
bonnet/0.34/+	server/0.39/+	passenger/0.44/+	diagnosis/0.30/+	hostel/0.34/+	protein/0.36/+	driver/0.30/+
bend/0.33/-	password/0.39/+	throat/0.43/-	bed/0.28/+	graduation/0.32/+	flora/0.36/-	vehicle body /0.30/+
tire/0.33/+	cartridge/0.38/+	sail/0.40/+	surgery/0.28/+	show/0.31/+	leaf/0.36/+	receipt/0.30/+
model/0.33/+	chip/0.38/+	captain/0.39/+	nurse/0.27/+	scholarship/0.31/+	agriculture/0.35/-	cabin/0.29/+
fuel/0.32/+	automation/0.38/+	rudder/0.38/+	chief-doctor /0.27/+	priestess/0.31/-	biology/0.35/-	wheel/0.28/+

steamship/0.32/-	interface/0.37/+	transportation /0.38/-	general director /0.27/+	dormitory/0.30 /+	medicine/0.35 /+	brake/0.28/+
dog/0.31/-	machine/0.37/-	chief officer /0.34/+	dispatch/0.26/ +	library/0.29/+	taste/0.34/+	free pass/0.28/+
gasoline/0.31/ +	computer virus /0.37/+	rope/0.33/+	tent/0.26/-	toilet/0.28/+	tea/0.33/-	fuel/0.28/+

Table 17. Various target antonyms, candidate antonyms, word2vec scores, class

Target antonym/candidate antonym/word2vec score/class(+ : correct antonym, - : wrong antonym)						
solution	**old**	**foreign**	**import**	**male**	**debt**	**peace**
formula/0.45/-	new/0.38/+	domestic/0.54/+	export/0.74/+	female/0.65/+	interest/0.43/-	stability/0.47/-
resort/0.43/-	communist/0.28/-	american/0.31/-	production/0.53/-	lady/0.57/+	credit/0.41/-	serenity/0.44/-
approach/0.36/-	currently/0.28/-	non-muslim/0.30/-	exporter/0.44/-	girl/0.41/+	installment/0.34/-	truce/0.43/-
peace/0.30/-	manager/0.24/-	racist/0.29/-	current/0.42/-	marriage/0.36/-	holding/0.33/-	friendship/0.42/-
disease	**near**	**defense**	**special**	**easy**	**compound**	**poor**
dementia/0.55/-	far/0.31/+	assault/0.37/+	public/0.26/+	possible/0.53/-	simple/0.43/+	indigent/0.79/-
infectious/0.47/-	neighbor/0.30/-	military/0.31/-	different/0.26/-	hard/0.50/+	element/0.25/-	rich/0.57/+
germ/0.47/-	brother/0.24/-	finance/0.31/-	gizli/0.26/-	comfortable/0.43/-	expression/0.16/-	poverty/0.56/-
health/0.45/+	belong/0.23/-	attack/0.31/+	separate/0.26/-	quickly/0.42/-	mixed/0.13/-	beggarly/0.49/-
real	**open**	**minority**	**amateur**	**win**	**strong**	**negative**
falsehood/0.45/+	closed/0.46/+	ethnic/0.43/-	professional/0.62/	defeat/0.68/+	weak/0.51/+	positive/0.67/+
reality/0.42/-	hidden/0.35/+	community/0.37/-	league/0.24/-	beating/0.66/+	ambitious/0.40/-	stable/0.43/-
lie/0.41/+	unlock/0.29/-	immigrant/0.36/-	different/0.24/-	championship/0.60	sturdy/0.38/-	plus/0.37/+
truth/0.39/-	latent/0.28/+	majority/0.33/+	young/0.24/-	point/0.49/-	powerless/0.35/+	neutral/0.32/-
light	**vertical**	**ceiling**	**real**	**soft**	**thin**	**abstract**
heavy/0.49/+	horizontal/0.56/+	wall/0.41/-	falsehood/0.45/+	hard/0.47/+	plump/0.35/-	figurative/0.46/-
severe/0.37/+	transition/0.34/-	base/0.38/+	reality/0.42/-	flexible/0.43/-	elegant/0.31/-	picture/0.38/-
soft/0.32/-	italic/0.17/-	board/0.27/-	lie/0.41/+	aggressive/0.40/-	mercerized/0.30/-	concrete/0.37/+
hard/0.32/-	spine/0.17/-	roof/0.24/-	truth/0.39/-	sharp/0.37/-	nuance/0.28/-	surrealist/0.36/-
white	**low**	**start**	**clean**	**artificial**	**weak**	**rout**
grey/0.44/-	high/0.27/+	finish/0.27/+	dirty/0.47/+	imitation/0.57/-	powerful/0.59/+	disaster/0.38/-
brown/0.44/-	silent/0.21/-	milestone/0.26/-	filthy/0.36/+	synthetic/0.33/-	strong/0.51/+	victory/0.31/+
colored/0.36/-	flat/0.16/-	new/0.25/-	regular/0.30/-	herbal/0.23/-	powerless/0.49/-	-
olive drab/0.36/-	superior/0.16/-	birth/0.22/-	honesty/0.27/-	based on/0.21/-	bad/0.35/-	-

REFERENCES

[1] Miller, George A. "WordNet: a lexical database for English." Communications of the ACM 38.11 (1995): 39-41.

[2] Hearst, M. A., (1992). "Automatic Acquisition of Hyponyms from Large Text Corpora", Proceedings of the 14thInternational Conference on Computational Linguistics, 23-28 August 1992, Nantes, France, 539-545.

[3] Rydin, S., (2002). "Building a Hyponymy Lexicon with Hierarchical Structure", Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition, 7-12 June 2002, Philadelphia, 26-33.

[4] Ando, M., Sekine, S. ve Ishizaki, S., (2004). "Automatic Extraction of Hyponyms from Japanese Newspapers Using Lexico-syntactic Patterns", In IPSJ SIG Technical Report, (98):77-82.

[5] Snow, R., Jurafsky, D. ve Ng, A., (2005). "Learning Syntactic Patterns for Automatic Hypernym Discovery", Advances in Neural Information Processing Systems, 5-9 December 2005, Vancouver, British Columbia, Canada, 1297-1304.
[6] Ritter, A., Soderland, S. ve Etzioni, O., (2009). "What Is This, Anyway: Automatic Hypernym Discovery", The Association for the Advancement of Artificial Intelligence, 23-25 March 2009, Palo Alto, California, USA, 88-93.
[7] Yıldırım, S. ve Yıldız, T., (2012). "Corpus-Driven Hyponym Acquisition for Turkish Language", CICLing 13th International Conference on Intelligent Text Processig and Computational Linguistics, 2012.
[8] Sak, H., Güngör, T. ve Saraclar, M., (2008). "Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus", The 6th International Conference on Natural Language Processing, 27-27 August 2008, Gothenburg, Sweden, 417-427.
[9] Yıldırım, Savas, and Tugba Yıldız. "Automatic Extraction of Turkish Hypernym-Hyponym Pairs From Large Corpus." 24th International Conference on Computational Linguistics. 2012.
[10] Sahin, G., Diri, B., &Yildiz, T. Extraction of Turkish Semantic Relation Pairs using Corpus Analysis Tool. International Journal of Computer and Information Technology, 5.6 (2016).
[11] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. "Distributed representations of words and phrases and their compositionality", In Proc. Advances in Neural Information Processing Systems 26 3111–3119 (2013).
[12] Şahin, G., Diri, B., & Yıldız, T. (2016, May). A hybrid method for extracting Turkish part-whole relation pairs from corpus. In Signal Processing and Communication Application Conference (SIU), 2016 24th (pp. 1037-1040). IEEE.
[13] Yıldız, Tuğba, Banu Diri, and Savaş Yıldırım. "Acquisition of Turkish meronym based on classification of patterns." Pattern Analysis and Applications 19.2 (2016): 495-507.
[14] Pantel, P., Pennacchiotti, M.: Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL, pp. 113–120. ACL Press, Sydney (2006).
[15] Cover, T.M. and Thomas, J.A. 1991. Elements of Information Theory. John Wiley & Sons.
[16] http://trec.nist.gov/data/qa/t9_qadata.html
[17] Dekang Lin, Shaojun Zhao, Lijuan Qin, ve Ming Zhou (2003). "Identifying Synonyms among Distributionally Similar Words", Proceedings of IJCAI, Acapulco, Mexico.
[18] Turney, P.D. (2008). "A Uniform Approach to Analogies, Synonyms, Antonyms, and Associations", Proceedings of the 22nd International Conference on Computational Linguistics, Manchester, UK, 905-912.
[19] Lobanova, A., van der Kleij, T. ve J. Spenader (2010). "Defining Antonymy: A Corpus-Based Study of Opposites by LexicoSyntactic Patterns", International Journal of Lexicography, 23:19-53.
[20] Silke Scheible, Sabine Schulte im Walde, ve Sylvia Springorum (2013)."Uncovering Distributional Differences between Synonyms and Antonyms in a Word Space Model", IJCNLP, 489-497.
[21] Enrico Santus, Qin Lu, Alessandro Lenci, and Chu-Ren Huang (2014). "Unsupervised Antonym-Synonym Discrimination in Vector Space", In Proceedings of the First Italian Conference on Computational Linguistics.
[22] D.A. Cruse. Lexical Semantics. Cambridge University Press, Cambridge, UK, 1986.
[23] Evert, S. (2005). The Statistics of Word Cooccurrences: Word Pairs and Collocations. Dissertation, University of Stuttgart.
[24] L. Aldubayi. ve M. Al-Yahya, (2015). "Automatic Arabic Antonym Extraction Using a Corpus Analysis Tool", Journal of Theoretical and Applied Information Technology, January 2015.

Author

Gurkan Sahin is a Ph.D. student in the computer Engineering Department at Yildiz Technical University (YTU). He received his B.S. and M.Sc. degree in the computer engineering from YTU in 2013 and 2016, respectively. His research interest includes natural language processing and machine learning.