International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 05 – Issue 06, November 2016

Extraction of Turkish Semantic Relation Pairs using Corpus Analysis Tool

Gurkan Sahin* and Banu Diri
Computer Engineering
Yıldız Technical University
Istanbul, Turkey
*Email: gurkansahin08 [AT] gmail.com

Tugba Yildiz
Computer Engineering
Istanbul Bilgi University
Istanbul, Turkey

***Abstract*— In this study, we have developed a Turkish semantic relation extraction tool. The tool takes an unparsed corpus as input and gives hyponym, meronym and antonym words with their reliability scores as output for given target words. Corpus is parsed by Turkish morphological parser called Zemberek and word vectors are created by Word2Vec for each unique word in corpus. To extract relation patterns, hyponymy, holonymy, antonymy pairs called initial seeds are prepared then, all possible relation patterns are extracted using initial seeds. Reliability of patterns are calculated using corpus statistics and various association metrics. Reliable patterns are selected to extract new semantic pairs from parsed corpus. To determine correctness of extracted pairs, total pattern frequency, different pattern frequency and Word2Vec vector cosine similarity have been used. After experiments, we have obtained 83%, 63%-86%, and 85% average precisions for hyponymy, holonymy and antonymy relations, respectively.**

Keywords — hyponymy; holonymy; antonymy; Word2Vec; semantic relation; pattern-based approach.

I. INTRODUCTION

In linguistics, words are connected to each other with various semantic relationships. Hyponymy, hypernymy, meronymy, holonymy, antonymy can be given as example to the most well-known semantic relationships.

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 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, also are co-hyponyms of each other.

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’, ‘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.

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. The relationship 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 replace with each other in the pattern, like ‘neither big nor small’ and ‘neither small nor big’.

Automatically extraction of semantic relation pairs from various sources like corpus, dictionary definitions, web pages etc. is one of the popular topics in natural language processing (NLP). In this way, WordNet-like semantic dictionaries can be easily created without human help. In semantic dictionary, each word is connected to each other with various semantic relationships. WordNet is a large lexical database of English consisting of nouns, verbs, adjectives and adverbs. In WordNet, each word is represented by the synsets consisting of synonyms of the word. Each of 117.000 synsets is linked to other synsets by means of various conceptual relations like hyponymy, holonymy etc. WordNet is a useful tool for computational linguistics and NLP applications like automatic question answering, information extraction etc.

In the literature, pattern-based methods are used usually to extract semantic pairs. Using a small number of patterns, semantic pairs can be easily obtained from given resources**.** In addition to patterns, part-of-speech tag is also used to obtain correct pairs. But, pattern-based method is not successful for all semantic relationships like synonymy, because there is no distinctive pattern for synonymy.

In this study, we aim to develop a NLP tool which can extract hyponymy, holonymy and antonymy pairs from given a Turkish corpus. The tool takes a corpus as input and given a target word, it extracts hyponym, meronym, antonym words as output according to different reliability scores.

Firstly, hyponym-hypernym, meronym-holonym and antonym pairs (initial seeds) are created. Using these seeds, hyponymy, holonymy and antonymy patterns are extracted from corpus. Then, reliable patterns which strongly represent relationship, are selected using dice, dice-idf, pmi, pmi-idf association measurement methods. Selected reliable patterns are used to extract new relational pairs from corpus.

It is necessary to determine whether new pairs are correct or not, because pairs which occur with reliable patterns may not be correct. To select correct pairs and eliminate wrong pairs, we need to use some scores which represent correctness probability of the pair. For this reason, co-occurrence frequency of pairs with all patterns, co-occurrence frequency of pairs with different patterns and Word2Vec vector similarity score are used.

In Chapter II, we have been mentioned about related studies. Used resources and tools have been given in Chapter III. Proposed method and experimental results have been given in Chapter IV and V, respectively. Finally, we have mentioned about future studies and have made general assessments in Chapter VI.

II. Related Studies

Hearst [1] shown that hyponym-hypernym pairs could be extracted easily from corpus with high accuracy using only a handful of pattern. Snow [2] used dependency patterns as feature to classify hyponym-hypernym pairs and best classification accuracy was obtained by logistic regression algorithm with 54%. Ando [3] extracted noun-noun type hyponym-hypernym pairs from Japanese newspaper archive. 30 candidate patterns were generated using initial seeds and only 7 high frequent patterns were used. After experiments, 48%-87% accuracy was obtained for 130 target hypernym. Rydin [4] created a hierarchical IsA (hyponymy) structure using Swedish newspaper corpus. Sang [5] classified hyponym-hypernym pairs using 16.728 patterns as feature and obtained 54% accuracy. Ritter [6] used pair-pattern co-occurrence frequency to classify hyponym-hypernym pairs. Hidden Markov Model (HMM) was used to identify IsA pairs which do not occur with patterns and recall increased from 80% to 82%. For Turkish [7], [8] studies were done.

Ittoo [9] et al. extracted meronym-holonym pairs from text database. Firstly, initial seed were created and using these seeds, all patterns were extracted from parsed database. To determine reliability of patterns, pointwise mutual information (pmi) association measurement was used. Selected reliable patterns were used to extract new meronym-holonym pairs and after experiments, 81% accuracy was obtained. Van Hage [10] et al. used 501 meronym-holonym pairs to extract patterns which will be used to generate new pairs. In this study, web pages and Google queries were used to obtain patterns and new pairs. Yıldız T. [11] extracted Turkish meronym-holonym pairs using Bootstrapped patterns (BP) method. In this method, patterns were generated using initial seeds and 64%-72% average accuracy was obtained for given target holonyms. Also, in [12] patterns and Word2Vec vector similarity score were used to extract Turkish meronym-holonym pairs.

Lobanova [13] worked on antonymy relation. Firstly, antonym patterns were generated from Dutch corpus using adjective-adjective antonym initial seeds. A reliability score was assigned to each pattern and reliable patterns were selected to generate new antonym pairs. Contrary to expectations, the majority of new extracted pairs were noun-noun type rather than adjective-adjective. Using initial seeds, antonym patterns were generated and new pairs were extracted using these patterns. This process continued throughout sixth iteration. At the end of sixth iteration, 28.3% and 45.4% accuracy rates were obtained for reliability scores of pairs >= 0.6 and >=0.9, respectively. Lin [14] used patterns of incompatibility to distinguish antonym from synonyms. Lin said that if any pair occurs with "from X to Y" or "either X or Y" patterns, this pair is semantically incompatible with each other. To distinguish antonym from synonyms, Lin used co-occurrence frequency between pairs with incompatible patterns in Web pages. To measure the success of method, 80 antonym and 80 synonym pair were selected, 86.4% precision and 95% recall were obtained. Turney [15] classified analogous pairs using corpus based supervised machine learning algorithm and used pair-pattern co-occurrence frequency as features. Totally 2.720 features were used and synonym and antonym pairs were classified with 75% accuracy using SVM algorithm. Santus [16] proposed APAnt (Average-Precision-Based) method to classify synonyms and antonyms. The method claims that synonym pairs occur with much more joint context words than antonyms. While high APAnt value represents high degree of antonym, low value represents low degree of antonym. To measure success of method, 2.232 test pair consisting of 1.070 antonym and 1.162 synonym pairs were created. Most related 100 context words were extracted using LMI (Local Mutual Information) score and average APAnt scores were calculated for two test groups. Boxplot distributions were examined and it was shown that APAnt method is better than baseline co-occurrence hypothesis to separate synonyms from antonyms. Mohammad et. al (2013) [17] prepared three group, which consist of antonym pairs, synonym pairs and random pairs. Similarity of each pairs were calculated using Lin's similarity formula (1998) [18] and corpus statistics. After experiments, it was shown that mean similarity of antonym and synonym pairs are greater than the random pairs. Surprisingly, it was shown that mean antonym similarity score is greater than mean synonym similarity score. The result gave information about the difficulty of separating antonym and synonym from each other. Unlike known, Schulte [19] showed that synonyms and antonyms can be distinguished from each other using useful context words. Synonyms and antonyms classified with 70.6% accuracy compared to 50% baseline accuracy. Luluh Aldhubayi and Maha Alyanya (2014) [20] classified Arabic antonym pairs using pair-pattern co-occurrence frequency and dice score. Noun-noun type antonym pairs were classified with 76% accuracy. In [21] Turkish antonyms were classified with 77.2% precision.

III. USED RESOURCES

Corpus: In this study, we used Turkish news text corpus [22] consisting of about 14 million web news. Turkish language is agglutinative language and is a member of the family of Altay language. Firstly, all words in corpus are parsed into roots and suffixes using Turkish Zemberek morphological parser [23] written in Java. The parser can generate multiple parsed results for every word but in this study, only first result is used. Because of the too large corpus, searching can be very time consuming. To accelerate searching process, parsed corpus is indexed by Apache Lucene 4.2.0 search library, which supports many searching operations such as term, phrase, regex, proximity, boolean query etc.

Word2Vec: Word2Vec [24] was created by a team of researchers at Google using C programming language. Word2Vec takes a corpus as input and generates high dimensionality word vectors as output for each unique words in corpus. Each word is clustered by Word2Vec according to context words and similar words are assigned to close coordinates. Word2Vec uses 2 different architecture called continuous bag-of-words (CBOW) and Skip-gram to produce distributed representation of words. CBOW architecture estimates a word by looking at the words around within a certain window size. In CBOW model, the order of context words does not influence bag-of-words assumption. The Skip-gram architecture model uses current word to predict surrounding window of context words [25]. In this method, closer context words have more weight than distant context words. Also in [24], it is said that CBOW is faster than Skip-gram model but, Skip-gram especially a better method for infrequent words. CBOW and Skip-gram architectures are given in Figure 1.

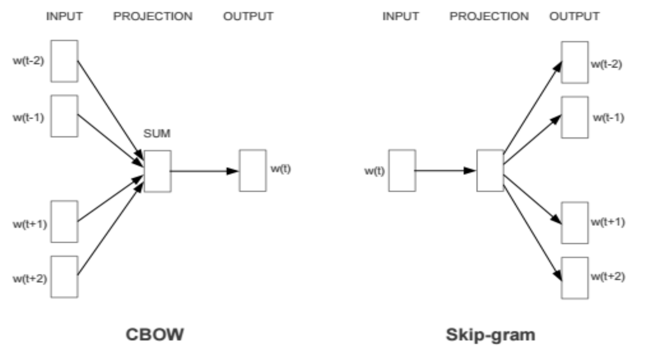


Figure 1. CBOW and Skip-gram architectures [26]

Apart from architecture type, performance of Word2Vec varies depending on several parameters. Word2Vec uses hierarchical softmax (default) or negative sampling algorithms for training. Dimension is important parameter, which determines word embedding will be represented with how many dimensions in word space. The representation quality of word embedding is increased to a degree with increasing size of representation dimensionality and then, it decreases after reaching a point. Default value of vector dimension is 100. Context window size determines how many context words will be evaluated before and after given a word. According to the authors' note, this parameter is 5 for Skip-Gram and 10 for CBOW architecture. Word2Vec also supports sub-sampling to select words, which have frequency above a certain value and to decrease training time. Word2Vec vector arithmetic could be used to extract word analogies using arithmetic operations such as "vec(king) - vec(man) = vec(queen) -vec(woman)".

IV. PROPOSED METHOD

A. *Extracting patterns from initial seeds*

Firstly, we prepare hyponym-hypernym, meronym-holonym and antonym pairs called initial seeds. These seeds are searched in the parsed corpus and related sentences are found. For each relationship, all possible patterns are extracted. Pattern structure is given below.

[0-1 WORD] X [0-3 WORD] Y [0-1 WORD]

In the pattern, X and Y represents hyponym-hypernym, meronym-holonym and antonym initial pairs. Maximum 3 words between pairs, maximum 1 word before X and maximum 1 word after Y are taken as pattern.

X-Y for hyponymy = (cat-animal) or (animal-cat)
X-Y for holonmy = (wheel-car) or (car-wheel)
X-Y for antonymy = (long-short) or (short-long)

B. *Selecting reliable patterns*

Although lots of patterns are extracted using initial seeds, all of these patterns may not represent semantic relations. Dice, dice-idf, pmi and pmi-idf association measurements are used to eliminate wrong patterns and select correct patterns. A reliability score is assigned for each extracted pattern using (1), (2), (3), and (4).

$$r(p) = \frac{\sum_{i\epsilon I}\left(\frac{dice_{(i,p)}}{max_{dice}} \text{ x } r(i)\right)}{|P|}; \qquad dice_{(i,p)} = \frac{2 \text{ x } |X,p,Y|}{|X,*,Y| + |*,p,*|} \qquad (1)$$

$$r(p) = \frac{\sum_{i\epsilon I}\left(\frac{dice\_idf_{(i,p)}}{max_{dice\_idf}} \text{ x } r(i)\right)}{|P|}; \; dice\_idf_{(i,p)} = dice(i,p) \times \frac{|*,*,*|}{|X,p,Y|} \qquad (2)$$

$$r(p) = \frac{\sum_{i\epsilon I}\left(\frac{pmi_{(i,p)}}{max_{pmi}} \text{ x } r(i)\right)}{|P|}; \qquad pmi_{(i,p)} = \log\left(\frac{|X,p,Y| \times |*,*,*|}{|X,*,Y| \times |*,p,*|}\right) \qquad (3)$$

$$r(p) = \frac{\sum_{i\epsilon I}\left(\frac{pmi\_idf_{(i,p)}}{max_{pmi\_idf}} \text{ x } r(i)\right)}{|P|}; \qquad pmi\_idf_{(i,p)} = pmi_{(i,p)} \times \frac{|*,*,*|}{|X,p,Y|} \qquad (4)$$

In (1), p represents any pattern, r(p) represents reliability of pattern, i or (X,Y) represents any relational pairs (hyponym-hypernym, meronym-holonym, antonym), r(i) represents reliability of initial seed. Because of all of initial seeds are correct, r(i) value of each initial seeds is equal to 1. max_{dice} is maximum dice score between all pairs and all pattern in corpus. This parameter is used to normalize the reliability score. In $dice_{(i,p)}$, |X,p,Y| is co-occurrence frequency of pattern p with X-Y pair. Also, |X,*,Y| is co-occurrence frequency of X-Y pair with all patterns and |*,p,*| is co-occurrence frequency of p pattern with all pairs in corpus. |P| is total number of pattern, which extracted using initial seeds. In (3), to get rid of negative logarithmic value, |*,*,*| parameter, which represents co-occurrence frequency of all pairs with all patterns or 3-gram frequency, is used. All patterns are sorted by reliability scores and most reliable patterns are selected to generate new hyponym-hypernym, meronym-holonym, antonym pairs. Reliable relation patterns are given in Table 1, Table 2, Table 3, respectively. In these tables highest reliability scores are given in bold format.

Table 1. Hyponymy patterns and reliability scores

Turkish patterns	**English patterns**	**Corpus frequency**	**Total pair frequency**	**Different pair frequency**	**Dice score**	**Dice-idf score**	**Pmi score**	**Pmi-idf score**
X gibi Y	Y such as X	72.343	337	71	9,26	1,93	1,63	0,12
X gibi bir Y		8.350	6	6	1,31	0,93	0,15	0,002
X gibi birçok Y		532	8	7	19,45	12,51	6,09	0,10
X gibi bazı Y		516	6	5	16,19	10,29	5,29	0,085
X gibi çeşitli Y		453	7	6	20,13	12,81	6,82	0,114
X ve diğer Y	X and/or other Y	8.222	76	33	17,5	6,72	2,95	0,10
X veya diğer Y		629	5	4	7,56	4,87	1,51	0,023
X ve benzeri Y		3.528	76	33	**38,28**	**14,61**	**6,89**	**0,25**
X ve çeşitli Y		776	5	4	9,86	6,12	2,80	0,046

In Table 1, X and Y represent any hyponym and hypernym, respectively. Most reliable hyponymy pattern is "X ve benzeri Y" for all reliability scores. Corpus frequency represents total co-occurrence frequency of all noun-noun pairs with the related pattern. The most abundant hyponymy pattern in our corpus is "X gibi Y" pattern which totally occur 72.343 times with noun-noun type X-Y pairs. Total pair frequency is total co-occurrence frequency of the related pattern in corpus with all of the initial seeds. Different pair frequency represents that how many different initial seed is observed with the related pattern. "X ve diğer Y" and "X gibi Y" patterns occur with 33 and 71 different initial pairs, respectively. Dice, dice-idf, pmi and pmi-idf points are pattern reliability scores which are calculated based on (1), (2), (3), and (4).

Table 2. Holonymy patterns and reliability scores

Turkish patterns	**English patterns**	**Corpus frequency**	**Total pair frequency**	**Different pair frequency**	**Dice score**	**Dice-idf score**	**Pmi score**	**Pmi-idf score**
X in Y si	Y of the X	549.516	8.616	364	5,21	26,48	2,62	**16,06**
X Y si		11.841.159	25.079	476	0,71	3,33	0,57	3,30
Y si olan X	X with Y	44.369	185	78	1,24	8,06	0,61	4,08
Y li X		1.234.127	25.007	349	**6,79**	**28,22**	**2,63**	15,12
Y siz X	X without Y	170.163	1.914	151	3,75	19,33	1,29	8,15

In Table 2, X and Y represent noun-noun type holonym and meronym term, respectively. "Y li X" pattern is the most reliable pattern for all reliability scores except pmi-idf. This pattern occurs with 349 different initial seeds and occurs

25.007 times with all of the seeds. The least reliable pattern is “X Y si” for all reliability scores. The most abundant meronym-holonym pattern in our corpus is “X Y si” pattern which totally occurs 11.841.159 times with noun-noun type X-Y pairs.

Table 3. Antonymy pattern groups and reliability scores

Pattern group	Turkish patterns	English patterns	Corpus frequency	Total pair frequency	Different pair frequency	Dice score	Dice-idf score	Pmi score	Pmi-idf score
G1	X ve Y arasında X ve Y arasındaki	between X and Y	1.396	818	46	155,49	**13,78**	**7,63**	1,85
G2	ne X ne Y ne X nede Y ne X ne de Y	neither X nor Y	2.370	1.005	58	105,71	13,00	3,37	1,91
G3	X yada Y X ya da Y	X or Y	35.232	5.359	83	**210,68**	4,63	1,92	**3,61**
G4	X ‘den Y ye	from X to Y	79.363	6.584	49	133,23	1,35	0,57	3,55
G5	X mı Y mı X mi Y mi X mu Y mu X mü Y mü	Is it X or Y?	879	108	22	38,72	7,61	2,77	0,23
G6	bir X bir Y	a/an X a/an Y	4,251	418	35	48.28	5.73	0.96	0.38

Turkish parser can label some adjective words as noun. For this reason, in antonym patterns, X and Y represent all of noun-noun, adjective-adjective, noun-adjective, and adjective-noun type antonym pairs. Similar antonym patterns are grouped in one pattern as G1, G2, G3, G4, G5, and G6. According to reliability scores, while the most reliable antonym pattern group is G3 for dice and pmi-idf scores, G1 is the most reliable pattern group for dice-idf and pmi scores. Similarly, while the least reliable group is G4 for dice-idf and pmi, G5 is the least reliable group for dice and pmi-idf. The most abundant antonymy pattern group in our corpus is “G4”, which totally occurs 79.363 times with all X-Y pairs. Also, “G5” is the least abundant antonymy pattern which occurs 879 times with all pairs.

C. *Extracting new semantic pairs*

After selecting reliable patterns, these pattern are used to extract new hyponym-hypernym, meronym-holonym and antonym pairs from the parsed corpus. Given a target word (hypernym, holonym or antonym) is searched with reliable patterns and semantic equivalent words, which called candidates (hyponyms, meronyms, antonyms) are extracted. Example target words, patterns and candidates are given in Table 4.

Table 4. Example target words, patterns and candidates

Target word (Turkish / English /type of target)	Turkish pattern	English pattern	Candidate (Turkish / English / type of candidate)
meyve / fruit / hypernym	***muz*** ve benzeri meyveler	fruits such as ***banana***	muz / banana / hyponym
ev / house / holonym	evin ***bahçe***si	***garden*** of the house	bahçe / garden / meronym
güzel / beautiful / antonym	güzel ya da ***çirkin***	beautiful or ***ugly***	çirkin / ugly / antonym

D. *Evaluation of the correctness of new pairs*

Three different methods have been used to evaluate correctness of new pairs.

Total pattern co-occurrence frequency (TPF): We assume that if any pair occurs with high frequency with all patterns, this pair is more correct than low frequent others. According to the method, all extracted candidates are sorted according to their total pattern co-occurrence frequency. After sorting, precision of first k pairs is determined by human labor. Value of k is given in Chapter V.

Different pattern co-occurrence frequency (DPF): We assume that if any pair occurs with high frequency with different patterns, this pair is more correct than low frequent others. According to this method, all candidates are sorted according to the different pattern co-occurrence frequency. Then, precision of first k pairs is determined by human labor.

Word2Vec vector similarity score (W2VS): Word2Vec vector cosine similarity is used to rank candidates. Cosine similarity (5) is a method used to measure the similarity between two vectors. According to cosine similarity, small angle between two vectors represents that these vectors very similar to each other and large angle between two vectors represents that these vectors low similar to each other.

$$\text{similarity}_{(v1,v2)} = \cos(\theta) = \frac{\sum_{i=1}^{N} V_{1,i}\, V_{2,i}}{\sqrt{\sum_{i=1}^{N} V_{1,i}^2}\sqrt{\sum_{i=1}^{N} V_{2,i}^2}} \quad (5)$$

In (5), N is number of dimension that each word is represented by Word2Vec. In this study, N is selected as 200. $V_{1,i}$ represents word vector for target hypernym, holonym and antonym words and $V_{2,i}$ represents word vector for candidate hyponym, meronym and antonym words. Candidates are sorted according to Word2Vec similarity scores and precision of first k pairs is calculated by human labor.

V. EXPERIMENTAL RESULTS

To determine success of the proposed method, hypernym, holonym and antonym target words are selected and these words are given the system as input.

For hyponymy relation, 15 target hypernym words are prepared. Target words are searched with related patterns and candidate hyponyms are extracted. New candidates are evaluated according to three different methods given before.

In corpus, while some of target hypernyms can have many hyponyms, others can have a small number hyponyms. For example, while “fruit” hypernym occurs with 20 different hyponyms, “sport” hypernym can occur with 40 different hyponyms in corpus. For this reason, it is important to decide that how many candidates must be evaluated to calculate precision. To overcome this problem, we found number of hyponyms in corpus for each target hypernym. In the corpus, 38 different hyponyms are found for “sport” hypernym and 27 different hyponyms are found for “fish” hypernym. For example, we select first 38 and 27 candidate hyponyms to calculate precision for “sport” and “fish” hypernyms, respectively. Target hypernyms and precisions are given in Table 5.

Table 5. Target hypernyms and precisions

Turkish target hypernyms	English target hypernyms	Number of correct hyponyms in corpus	Number of extracted candidates	TPF precision	DPF precision	W2VS precision
Alet	Tool	71	82	67/71 = 0.94	67/71= 0.93	67/71= 0.93
Balık	Fish	27	40	21/27 = 0.78	21/27 = 0.78	24/27 = 0.89
Bilim	Science	29	49	22/29 = 0.76	22/29 = 0.76	23/29 = 0.79
Bitki	Plant	57	66	50/57 = 0.88	50/57 = 0.88	50/57 = 0.88
Cihaz	Device	35	57	30/35 = 0.86	30/35= 0.86	30/35= 0.86
Eşya	Goods	109	142	95/109 = 0.87	93/109 = 0.85	96/109 = 0.88
Görevli	Attendant	38	52	30/38= 0.79	31/38= 0.82	34/38= 0.89
Hayvan	Animal	77	103	65/75= 0.87	65/75= 0.87	64/75= 0.85
İçecek	Beverage	23	25	21/23= 0.91	21/23= 0.91	22/23= 0.96
Meslek	Job	88	141	60/88 = 0.68	60/88 = 0.68	68/80 = 0.85
Meyve	Fruit	26	47	20/26= 0.77	20/26= 0.77	20/26= 0.77
Mineral	Mineral	28	32	20/28= 0.71	20/28= 0.71	20/28= 0.71
Organ	Organ	27	48	19/27= 0.70	19/27= 0.70	13/27= 0.48
Sebze	Vegetable	29	35	28/29= 0.97	28/29= 0.97	25/29= 0.86
Spor	Sport	38	87	25/38= 0.66	25/38= 0.66	30/38= 0.79
Average				**0.81**	**0.81**	**0.83**

81%, 81%, 83% average precisions are obtained from TPF, DPF, and W2VS methods, respectively. It is shown that W2VS method is a bit more successful than other methods to select correct hyponym-hypernym pairs.

For holonymy relation, 18 target holonym words are prepared. Target words are searched with related patterns and candidate meronyms are extracted. New candidates are evaluated according to three different methods. As a result of experiments conducted, we observed that number of hyponyms for given target hypernyms are less than number of meronyms for given target holonyms, according to productivity of hyponymy and holonymy patterns. For this reason, first 10, 20, 30, 40, and 50 sorted candidate meronyms are selected to measure precisions.

Table 6. Target holonyms and precisions

Turkish target holonyms	English target holonyms	Number of extracted candidates	TPF precision (first 10-20-30-40-50 pairs)	DPF precision (first 10-20-30-40-50 pairs)	W2VS precision (first 10-20-30-40-50 pairs)
Ağaç	Tree	275	0.80-0.70-0.60-0.53-0.48	0.80-0.75-0.67-0.60-0.60	0.70-0.75-0.67-0.63-0.54
Araba	Car	310	0.60-0.50-0.60-0.60-0.64	1.0-0.75-0.77-0.70-0.72	1.0-0.80-0.83-0.83-0.74
Bilgisayar	Computer	568	0.70-0.70-0.57-0.53-0.46	0.90-0.85-0.80-0.73-0.58	0.90-0.85-0.83-0.83-0.80
Bitki	Plant	213	0.60-0.60-0.63-0.58-0.54	1.0-0.80-0.63-0.60-0.54	0.80-0.75-0.77-0.75-0.66
Ev	Home	656	0.50-0.55-0.53-0.45-0.46	1.0-0.85-0.70-0.63-0.60	0.40-0.55-0.53-0.53-0.54
Gemi	Ship	398	0.90-0.65-0.53-0.53-0.50	1.0-0.90-0.83-0.75-0.68	0.50-0.60-0.70-0.75-0.68
Hastane	Hospital	342	0.90-0.65-0.60-0.60-0.60	0.80-0.80-0.70-0.63-0.62	1.0-0.90-0.73-0.63-0.60
Hayvan	Animal	469	0.70-0.60-0.57-0.45-0.42	0.70-0.60-0.60-0.60-0.56	0.30-0.20-0.37-0.38-0.42
İnsan	Human	1326	0.50-0.60-0.53-0.53-0.58	0.60-0.75-0.80-0.78-0.78	0.30-0.45-0.50-0.55-0.58
Kitap	Book	565	0.70-0.60-0.67-0.63-0.60	0.90-0.85-0.73-0.65-0.54	0.60-0.75-0.57-0.65-0.62
Meyve	Fruit	231	0.40-0.40-0.47-0.48-0.46	1.0-0.80-0.77-0.63-0.54	0.20-0.35-0.40-0.40-0.42
Okul	School	636	0.80-0.70-0.67-0.65-0.58	0.80-0.70-0.80-0.73-0.72	0.80-0.85-0.83-0.85-0.78
Otobüs	Bus	246	0.90-0.60-0.67-0.65-0.62	1.0-0.95-0.83-0.73-0.68	0.70-0.80-0.70-0.73-0.70
Radyo	Radio	292	0.90-0.75-0.70-0.68-0.60	0.70-0.80-0.70-0.68-0.66	0.70-0.80-0.80-0.80-0.82
Telefon	Telephone	480	1.0-0.85-0.77-0.65-0.62	0.80-0.80-0.70-0.73-0.72	0.90-0.85-0.83-0.80-0.76
Televizyon	Television	477	0.80-0.60-0.60-0.55-0.52	0.80-0.60-0.50-0.50-0.48	0.90-0.95-0.93-0.90-0.80
Uçak	Airplane	393	0.60-0.55-0.60-0.58-0.56	1.0-0.90-0.83-0.78-0.70	0.90-0.85-0.80-0.70-0.66
Üniversite	University	545	0.90-0.60-0.60-0.63-0.54	0.70-0.70-0.67-0.65-0.66	0.80-0.90-0.93-0.80-0.68
Average			**0.73-0.62-0.61-0.57-0.54**	**0.86-0.79-0.72-0.67-0.63**	**0.69-0.72-0.71-0.70-0.66**

Average precisions are given in Chart 1 for holonymy. For antonymy relation, 80 target antonym words are prepared. Target words are searched with antonym patterns and candidate antonyms are extracted. New candidates are evaluated according to three different methods. A word can have more than one semantically opposite word. But, a word is usually used with the antonym word which has the most opposite meaning. For this reason, only first candidate is selected from the sorted candidates to calculate precision. The obtained average precisions are given in Chart 2. The highest average precision has been obtained from DPF method as 85%. Unlike experimental results of hyponymy and holonymy relation, W2VS has produced the lowest average precision for antonymy.

Chart 1. Average precisions for first 10, 20, 30, 40, and 50 meronym-holonym pairs according to 3 different methods

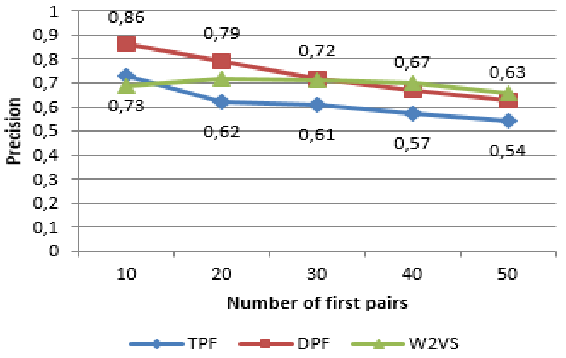
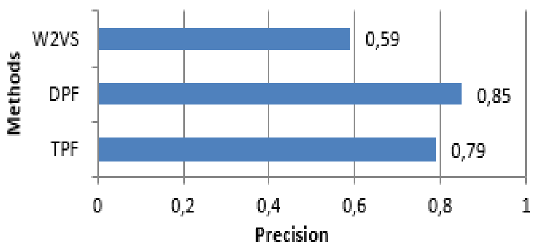


Chart 2. Average precisions for antonym pairs according to 3 different methods



VI. Conclusions

In this study, a Turkish NLP tool has been developed. The tool extracts hyponymy, holonymy and antonymy pairs from given a corpus. Firstly, hyponymy, holonymy, antonymy pairs are prepared. These seeds are searched in parsed corpus and all relation patterns are extracted. Reliability scores are assigned to each pattern. Reliable patterns are selected and these patterns are used to generate new relation pairs.

To test the system, 15 target hypernym, 18 target holonym and 80 target antonym words are prepared. These target words are searched with relevant patterns and candidate hyponyms, meronyms and antonyms are extracted from parsed corpus. To measure precision of candidates, total pattern frequency, different pattern frequency and Word2Vec vector similarity are used.

For hyponymy relation 81%, 81%, and 83% average precisions are obtained for target hypernyms using TPF, DPF, and W2VS scoring methods, respectively. For holonymy relation, the best average precision is obtained for DPF method within range 86%-63% for first 10 and 50 candidates. Also, the worst average precisions are obtained for TPF method for holonymy. For antonymy relation, the best average precision is obtained for DPF method as 85%. After experiments it is clearly shown that although Word2Vec vector similarity is used successfully to select correct hyponym-hypernym and meronym-holonym pairs, it is not successful to select correct antonym pairs.

Development of semantic pair extraction tool for Turkish is the biggest contribution of this study. Thanks to the tool, hyponymy, holonymy, antonymy relation pairs can be extracted easily from given an unparsed corpus. Using the Word2Vec vector similarity to select correct relation pairs is another contribution of this study.

In future studies, we aim to extract semantic pairs using Word2Vec arithmetic operations such as "vec(king) - vec(man) = vec(queen) - vec(woman)". Also, generate a web-based semantic pair extraction tool, which uses web pages, is aimed. In this way, extraction of so many relational pairs and creation of large lexical semantic dictionary for Turkish will be possible.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Sang, E.F.T.K. ve Hofmann, K., (2007). "Automatic Extraction of Dutch Hypernym-Hyponym Pairs", Proceedings of the 17th Meeting of Computational Linguistics, 12 January 2007, Netherlands.

[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] Şahin, G., Diri, B., & Yıldız, T. (2015, Mayıs). "Pattern and Semantic Similarity Based Automatic Extraction of Hyponym-Hypernym Relation from Turkish Corpus". 23nd Signal Processing and Communications Applications Conference (SIU), Malatya, (pp. 674 - 677). IEEE.

[9] Ittoo, A. ve Bouma, G., (2010). "On Learning Subtypes of The Part-whole Relation: Do Not Mix Your Seeds", Association for Computational Linguistics, 11-16 July 2010, Uppsala, Sweden.

[10] Van H., W.R., Kolb, H. ve Schreiber, G., (2006). "A Method for Learning Part-whole Relations", InternationalSemantic Web Conference, Lecture Notes in Computer Science, 4273:723-735.

[11] Yıldız, T., Yıldırım, S. ve Diri, B., (2013). "Extraction of Part-Whole Relations from Turkish Corpora", Computational Linguistics and Intelligent Text Processing, Greece.

[12] Şahin, G., Diri, B., & Yıldız, T. (2016, Mayıs). A Hybrid Method for Extracting Turkish Part-Whole Relation Pairs from Corpus. 24nd Signal Processing and Communications Applications Conference (SIU), Zonguldak.

[13] Lobanova, A., van der Kleij, T. ve J. Spenader (2010). "Defining Antonymy: A Corpus-Based Study of Opposites by Lexico-Syntactic Patterns", International Journal of Lexicography, 23:19-53.

[14] Dekang Lin, Shaojun Zhao, Lijuan Qin, ve Ming Zhou (2003). "Identifying Synonyms among Distributionally Similar Words", Proceedings of IJCAI, Acapulco, Mexico.

[15] 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.

[16] Enrico Santus, Qin Lu, Alessandro Lenci, ve Chu-Ren Huang (2014). "Unsupervised Antonym-Synonym Discrimination in Vector Space", In Proceedings of the First Italian Conference on Computational Linguistics.

[17] Mohammad, Saif, Bonnie Dorr, Graeme Hirst, ve Peter D. Turney (2013). "Computing Lexical Contrast", Computational Linguistics, 39(3):555–590.

[18] Dekang Lin. (1998). "Automatic Retrieval and Clustering of Similar Words", In Proceedings of COLING, Montreal, Canada.

[19] 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.

[20] 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.

[21] Şahin, G., Diri, B., & Yıldız, T. (2015). "Analysis of Lexico-syntactic Patterns for Antonym Pair Extractıon from a Turkish Corpus". Computer Science & Information Technology, 11-17.

[22] 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.

[23] Ahmet Afsin Akın and Mehmet Dundar Akın. Zemberek, an open source nlp framework for turkic languages. Structure, 2007.

[24] 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).

[25] https://en.wikipedia.org/wiki/Word2vec

[26] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space". CoRR, abs/1301.3781, 2013.