International Journal of Computer Engineering and Information Technology

VOL. 8, NO. 10, October 2016, 196–202
Available online at: www.ijceit.org
E-ISSN 2412-8856 (Online)

Classification of Turkish Semantic Relation Pairs Using Different Sources

Gürkan ŞAHİN

Computer Engineering, Yıldız Technical University, Istanbul, Turkey.

gurkansahin08@gmail.com

ABSTRACT

In this study, Turkish hyponymy, holonymy and antonymy semantic pairs are classified by machine learning algorithms. To classify different semantic pairs, lexico-syntactic patterns which obtained from large Turkish corpus, WordNet similarity scores and Word2Vec vector similarity are used. Each feature set is used individually to examine effects on classification accuracy. After experiments, it is shown that hyponymy, holonymy and antonymy pairs can be distinguished from each other using important features. Best classification result has been obtained using lexico-syntactic pattern features from random forest algorithm as 84% F score. Adding WordNet and Word2Vec features to lexico-syntactic patterns has not increased classification success significantly.

Keywords: *Hyponymy, Hypernymy, Holonymy, Meronymy, Antonymy, Semantic Relation, Word2Vec, WordNet, Machine Learning, Lexico-Syntactic Patterns, Classification.*

1. 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 be replaced 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 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, hyponymy, holonymy, antonymy pairs have been classified by machine learning algorithms. Lexico-syntactic patterns which extracted from parsed corpus, WordNet similarity scores, and Word2Vec vector similarity are used as three different feature set.

Section 2 mentions about past studies. In section 3, used resources are given. Section 4 describes features which will be used to classification. In section 5, experimental result is given. Finally, general assessment is given in Section 6.

2. RECENT 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 was 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] various association metrics 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 co-occur with “from X to Y” and “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 score are 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 then, 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.

3. USED RESOURCES

In this study, different resources have been used. These resources are given below, respectively.

3.1 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.

3.2 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 word 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.

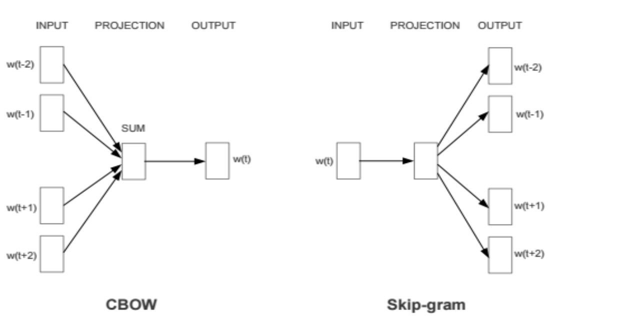

Fig. 1. CBOW and Skip-gram architectures [26]

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)”.

3.3 WordNet

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 in WordNet connected 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.

There are eight different semantic similarity measurement method in WordNet. Although each of these methods has different formulas, all of them use IsA hierarchy in WordNet. These methods are given below.

Hso (Hirst & St-Onge): The method was developed by Hirst and St-Onge [27] in 1998.

Jcn (Jiang-Conrath): Jiang and Conrath proposed this method to calculate semantic relatedness between two concepts in 1997 [28]. This method takes advantage of the number of nodes nodes in WordNet IsA hierarchy.

Lch (Leacock & Chodorow): This formula was presented by Leacock and Chodorow in 1998 [29]. This method (1) uses the distance and depth of information between word meanings.

$$\mathrm{lch}(c1, c2) = -\log\left(\frac{\mathrm{length}(c1, c2)}{2D}\right) \tag{1}$$

The method uses IsA hierarchy as path information. In (1), length(c1, c2) is number of nodes in shortest path between c1 and c2 concepts. D is maximum deep of IsA relation in WordNet.

Wup (Wu & Palmer): The method (2) was proposed by Wu and Palmer in 1994 [30].

$$\mathrm{wup}(c1, c2) = \frac{\mathrm{depth}(\mathrm{LCS}(C1, C2))}{\mathrm{depth}(c1) + \mathrm{depth}(c2)} \tag{2}$$

Depth(LCS(c1,c2)) represents the closest distance from closest ancestor of c1 and c2 (Least Common Subsumer (LCS)) to root node in the hierarchy. depth(c1) is number of nodes between c1 and root node. Also, depth(c2) is number of nodes between c2 and root node.

Resnikvector: The method (3) was proposed by Resnik in 1995 [31]. The method uses corpus statistics (one-million-word Brown Corpus of American English -Francis and Kucˇera 1982) [32] to measure similarity between two concepts. Similarity is calculated based on information content (ic). The information content can be explained briefly as the number of binary decisions required to find the information.

$$res(c1,c2) = ic(LCS(c1,c2)) \tag{3}$$

Lesk: The method was proposed by Lesk in 1986 [33]. Method uses gloss overlaps to measure similarity between concepts. Banerjee ve Pedersen (2002) applied this method to WordNet, later.

Lin: The method was proposed by Lin (1998) [34] to measure semantic similarity between two concepts.

Path: The method uses shortest path information between two concepts in IsA hierarchy. Semantic similarity between two concepts is calculated based on number of nodes in the shortest path. If the path has more nodes, then these two concepts are less similar to each other and vice versa. The similarity between two concepts is calculated based on inverse node number in shortest path [35]. If two concepts are same, then similarity is equal to 1.

To classify hyponymy, holonymy and antonymy pairs, these WordNet scores are used as features. To use similarity functions, WS4J [36] java library is used. All scores are normalized between [0-1] range before using, because each similarity method produces scores within different range.

3.4 Tureng Online Dictionary

To use WordNet similarity functions, firstly Turkish concepts are translated to English. For this task, Tureng online dictionary is used. There may be more than one translated candidates, but we use only first result as correct.

3.5 Apache Lucene

Lucene is a search engine which developed in Java language. Lucene supports many searching operations such as term, phrase, regex, proximity, boolean query etc. Firstly, parsed corpus is indexed by Lucene 4.2.0, then the index file is used to searching process.

4. FEATURE EXTRACTION

Three different feature groups are used to classification task. First group consists of lexico-syntactic patterns. Second group is WordNet similarity scores. Finally, third group is Word2Vec vector similarity score.

4.1 Lexico-Syntactic Pattern Features

Lexico-syntactic patterns are used to classify three different relational pairs. The following processing steps are applied to extract semantic relation patterns.

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 represent 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 holonymy = (wheel-car) or (car-wheel)
X-Y for antonymy = (long-short) or (short-long)

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 (4), (5), (6), and (7).

$$r(p) = \frac{\sum_{i \in I}\left(\frac{dice_{(i,p)}}{max_{dice}} \times r(i)\right)}{|P|}; \qquad dice_{(i,p)} = \frac{2 \times |X,p,Y|}{|X,*,Y| + |*,p,*|} \tag{4}$$

$$r(p) = \frac{\sum_{i \in I}\left(\frac{dice\_idf_{(i,p)}}{max_{dice\_idf}} \times r(i)\right)}{|P|}; \quad dice\_idf_{(i,p)} = dice(i,p) \times \frac{|*,*,*|}{|X,p,Y|} \tag{5}$$

$$r(p) = \frac{\sum_{i \in I}\left(\frac{pmi_{(i,p)}}{max_{pmi}} \times r(i)\right)}{|P|}; \qquad pmi_{(i,p)} = \log\left(\frac{|X,p,Y| \times |*,*,*|}{|X,*,Y| \times |*,p,*|}\right) \tag{6}$$

$$r(p) = \frac{\sum_{i \in I}\left(\frac{pmi\_idf_{(i,p)}}{max_{pmi\_idf}} \times r(i)\right)}{|P|}; \quad pmi\_idf_{(i,p)} = pmi_{(i,p)} \times \frac{|*,*,*|}{|X,p,Y|} \tag{7}$$

In (4), 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 (6), 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.

Table 1: Hyponymy patterns and reliability score

Turkish patterns	English patterns	Corpus frequency	Dice score
X gibi Y	Y such as X	72.343	9,26
X gibi bir Y		8.350	1,31
X gibi birçok Y		532	19,45
X gibi bazı Y		516	16,19
X gibi çeşitli Y		453	20,13
X ve diğer Y	X and/or other Y	8.222	17,5
X veya diğer Y		629	7,56
X ve benzeri Y		3.528	38,28
X ve çeşitli Y		776	9,86

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. Dice score is pattern reliability score which is calculated based on (4).

Table 2: Holonymy patterns and reliability score

Turkish patterns	English patterns	Corpus frequency	Dice score
X in Y si	Y of the X	549.516	5,21
X Y si		11.841.159	0,71
Y si olan X	X with Y	44.369	1,24
Y li X		1.234.127	6,79
Y siz X	X without Y	170.163	3,75

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. 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 occur 11.841.159 times with noun-noun type X-Y pairs.

Table 3: Antonymy patterns and reliability score

Pattern group	Turkish patterns	English patterns	Corpus frequency	Dice score
G1	-X ve Y arasında -X ve Y arasındaki	between X and Y	1.396	155,49
G2	-ne X ne Y -ne X nede Y -ne X ne de Y	neither X nor Y	2.370	105,71
G3	-X yada Y -X ya da Y	X or Y	35.232	210,68
G4	-X 'den Y ye	from X to Y	79.363	133,23
G5	-X mı Y mı -X mi Y mi -X mu Y mu -X mü Y mü	Is it X or Y?	879	38,72
G6	-bir X bir Y	a/an X a/an Y	4,251	48.28

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 occur 79.363 times with all X-Y pairs. Also, "G5" is the least abundant antonymy pattern which occurs 879 times with all pairs.

4.2 WordNet Features

8 different WordNet similarity methods are used as features. Firstly, pairs are given to Tureng and English equivalents are obtained. Then, WordNet similarities are calculated for each pair. Similarity methods produce scores within different ranges. Thus, all output values of all methods are normalized within range from 0 to 1.

Table 4: Some pairs and WordNet similarity scores

Turkish pair	English pair	HSO	LCH	LIN	PATH	WUP
motor-araba	engine-car	0,37	0,35	0,43	0,09	0,58
senato-üniversite	senate-college	0,0	0,47	0,40	0,14	0,62

4.3 Word2Vec Feature

Each word in the corpus is represented as 200-dimensional vector. To measure Word2Vec similarity between two words, cosine similarity (8) has been used and this similarity score used as Word2Vec similarity feature.

$$\text{similarity}_{(v1,v2)} = \cos(\theta) = \frac{\sum_{i=1}^{N} V_{1,i}\, V_{2,i}}{\sqrt{\sum_{i=1}^{N} V_{1,i}^2}\sqrt{\sum_{i=1}^{N} V_{2,i}^2}} \quad (8)$$

In (8), 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.

5. EXPERIMENTAL RESULTS

To test the proposed method, we have to prepare sample pairs. Hyponym-hypernym, meronym-holonym and antonym pairs are created. Then, these pairs are searched within 10 words window and co-occurrence frequency of pairs is calculated. All groups of semantic pairs are sorted by co-occurrence frequency and first 100 pairs are selected for each semantic relation.

Each pair is represented in terms of features. For classification task, Weka tool is used. For all classification algorithms, 10 fold cross validation has been applied to dataset.

Table 5: Feature groups

Feature group	Feature name	Number of features
F1	Lexico-syntactic patterns	20
F2	Word2Vec vector similarity	1
F3	WordNet similarities	8

The impact of the success of each feature group on classification is examined individually.

Table 6: Classification results for feature groups

	F scores of feature groups			
Algorithm	**F1**	**F2**	**F3**	**F1+F2+F3**
Naïve bayes	0.68	**0.44**	0.48	0.72
Logistic regression	0.80	0.36	0.47	0.81
Multilayer perceptron	0.75	0.41	0.50	0.74
Bagging	0.81	0.40	0.56	0.82
Random subspaces	0.80	0.41	0.55	0.81
Rotation forest	0.80	0.31	0.56	**0.85**
Random forest	**0.84**	0.42	**0.58**	0.84
J48	0.80	0.31	0.54	0.81
kNN	0.80	0.40	0.48	0.60
SVM	0.54	0.43	0.45	0.48

As a result of this study, we have obtained the highest classification success as 84% from random forest algorithm using lexico-syntactic pattern features. Using only WordNet similarity scores, 58% F score is obtained against 33.3% baseline accuracy. The lowest classification success is obtained as 44% from Naïve bayes algorithm using Word2Vec similarity feature. Finally, 85% classification success is obtained from rotation forest algorithm using combination of all the features.

6. CONCLUSIONS

In this study, we aim to classify hyponymy, meronymy and antonymy pairs using features which collected from three different sources. Lexico-syntactic patterns are extracted from parsed Turkish corpus and are used as feature set. Also, WordNet similarity scores and Word2Vec vector similarity score are used as feature set.
To test the system, 300 pairs consisting of 100 hyponym-hypernym, 100 meronym-holonym and 100 antonym pairs are created. Different machine learning algorithms and 10 fold-cross validation are applied to pairs. Also, each set of features is used individually to measure the success of the classification. After experiments, best classification result is obtained from lexico-syntactic pattern features as 84%. 58% and 44% F scores are obtained from WordNet features and Word2Vec feature, respectively.
In this study, we have shown that Turkish hyponymy, holonymy and antonymy pairs can be distinguished from each other successfully using significant features. Also, apart from lexico-syntactic patterns, we have examined effects of WordNet and Word2Vec on classification of semantic pairs. As a result, it is clearly shown that lexico-syntactic pattern features are more successful to classify semantic pairs against WordNet and Word2Vec features. This is another contribution of the study.

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", International Semantic 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.

[27] Hirst, G. ve St-Onge, D., (1998). "Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms", 305-332, Derleyen: Christiane Fellbaum, (1998). WordNet: An Electronic Lexical Database, MIT Press, Cambridge.

[28] Jiang, J. ve Conrath, D., (1997). "Semantic Similarity based on Corpus Statistics and Lexical Taxonomy", Proceedings of International Conference on Research in Computational Linguistics, 22-24 August 1997, Taipei, Taiwan, 19-33.

[29] Leacock, C. ve Chodorow, M., (1998). "Combining Local Context and WordNet Similarity for Word Sense Identification", 265-283; Derleyen: Christiane Fellbaum (1998). WordNet: An Electronic Lexical Database, MIT Press, Cambridge.

[30] Wu, Z. ve Palmer, M., (1994). "Verb Semantics and Lexical Selection", 32nd Annual Meeting of the Association for Computational Linguistics, 27-30 June 1994, Las Cruces, New Mexico, USA, 133-138.

[31] Resnik, P., (1995). "Using Information Content to Evaluate Semantic Similarity", Proceedings of the 14thInternational Joint Conference on Artificial Intelligence, IJCAI 95, 20-25 August 1995, Montréal Québec, Canada, 448-453.

[32] Wikipedia, Brown Corpus, https://en.wikipedia.org/wiki/Brown_Corpus, 21 Ocak 2016.

[33] Banerjee, S. ve Pedersen, T., (2002). "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet", Computational Linguistics and Intelligent Text Processing, 17-23 February 2002, Mexico City, Mexico.

[34] Lin, D., (1998). "An Information-Theoretic Definition of Similarity", Proceedings of the 15th International Conference on Machine Learning, 24-27 July 1998, Madison, Wisconsin, USA, 296-304.

[35] Cpan, Text - NSP, http://search.cpan.org/~tpederse/ WordNetSimilarity/lib/ WordNet/Similarity/path.pm, 22 Ocak 2016.

[36] WS4J Demo, WS4J, https://code.google.com/archive/p/ws4j/.