Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol. 9, No. 1, February 2021, pp.184-193

AWN-similarity: Towards developing free open-source frameworks for measuring Arabic semantic similarity under Windows / Linux operating systems

Faaza A. Almarsoomi[1], Israa A. Alwan[1]
[1] Department of Computer Science, Faculty of Education for Pure Sciences Ibn Al-Haitham, University of Baghdad

ABSTRACT

Arabic is a highly systematic language where its words exhibit elegant and rigorous logic. The field of Arabic word semantic similarity becomes more challenging due to its higher complexity and subtlety. This research is concerned with investigating the development of free open-source frameworks containing packages to calculate the semantic similarity between two Arabic words or concepts. These packages are known as AWN-ConceptSimilarity and AWN-WordSimilarity. The developed packages implement seven semantic similarity algorithms. One of these algorithms was proposed for Arabic and the rest were proposed for English where successfully adapted to Arabic using an Arabic lexical database, Arabic wordnet.

The functionality of the developed packages is validated using two-word similarity benchmarks datasets previously produced for Arabic. The results of the validation process indicate that the developed frameworks represent an important contribution to the Arabic semantic similarity field. Moreover, the developed packages are reliable to use and embed them with Arabic researchers' projects for improving or comparing their methodologies.

Keywords: Semantic similarity, Open-source framework, Arabic word similarity, Arabic WordNet.

Corresponding Author:

Faaza A. Almarsoomi,
Department of Computer Science, Faculty of Education for Pure Sciences ibn Al-Haitham,
University of Baghdad, Iraq
E-mail: faaza.a.a@ihcoedu.uobaghdad.edu.iq

1. Introduction

Evaluation of semantic similarity through the use of network representations has for a long time been gained attention from experts in the fields of psychology and artificial intelligence. Semantic word similarity algorithms identify how similar a pair of words is by determining the closeness of their concepts in a hierarchy. Various researchers have presented many computational techniques of semantic similarity dating back to Quillian, 1968 [1] and his approach of spreading activation. The proposed techniques try to emulate human ability as closely as possible resulting in improvements in a wide range of real live applications such as text mining, essay evaluation, dialogue systems, image retrieval from the Web, word sense disambiguation, machine translation, ontology mapping, and Web page retrieval, to name just a few.

An interesting challenge is that the proposed techniques are independent of each other and depend on different standards and programming languages. With these issues, it is difficult to implement the proposed techniques consistently and systematically compare the outcomes given by various algorithms. This problem was addressed by developing open-source frameworks containing packages to calculate the similarity score using a number of previously proposed techniques and also provide an interface that allows interaction with ontological resource. The most popular software package was developed by Pedersen et al. 2004 [2] and known as ***WordNet:Similarity***. It is a freely available for researcher which implements six similarity measures and three relatedness measures, all of which are on the basis of an English knowledge source, WordNet [3]. The

implemented measures were written as Perl modules for Linux OS users which take two concepts as input and give a score as a degree of similarity or related. The developed package can be embedded within researchers Perl programs as a module and calling its algorithms. The developed framework has been employed by other researchers for improving or comparing their methodology [4], [5], [6], and [7].
Java WordNet Similarity Library (JWSL) developed to provide researchers with a tool for accessing the English wordnet. It is a set of Java modules that implements several semantic similarity measures and can be extended by adding new algorithms.
UMLS-Similarity is an open-source framework developed by ***McInnes et al. 2009*** [8] for measuring the similarity between concepts belongs to the biomedical field. This framework was written in Perl and presented on the basis of the Unified Medical Language System (UMLS) and determined the similarity by adapting the previously developed similarity algorithms. It was created with the ability to add new similarity algorithms.

As can be noticed, the developed packages were produced to implement similarity measures for English language. To the best of our knowledge, there are no such packages developed for Arabic language. This research is concerned with investigating the development of free open-source frameworks containing packages to calculate the semantic similarity for Arabic language which can be used by Arabic researchers for improving and comparing their methodologies. The developed packages will be presented as sets of Java and Perl modules for Linux/Windows OS users. Very few works have been presented in the field of Arabic word semantic similarity. Consequently, the first step in the process of the development of these packages is adapting the English semantic similarity algorithms to Arabic. All implemented algorithms in English packages rely on English wordnet. Therefore, this paper will exploit an Arabic knowledge source known as Arabic wordnet (AWN) [9] for calculating the similarity where the AWN creation methodology was on the basis of the design and contents of English WordNet. Seven similarity algorithms will be implemented to identify the similarity between two Arabic concepts. These algorithms will be included in a package known as AWN-ConceptSimilarty. This package can be used with applications that have sense-tagged data such as in word sense disambiguation for determining the correct sense that should be used in a particular context where the sense with a highest similarity score is the one being used. For applications that do not have sense tagged data, the adapted similarity algorithms are implemented to determine the similarity between two Arabic words rather than two concepts. These algorithms will be included in a package known as AWN-WordSimilarity. The functionality of the developed packages is validated using two-word similarity benchmarks datasets previously produced for Arabic known as ANSS-70 and Evaluation datasets.

The rest of this paper includes reviewing the ontology-based similarity algorithms that supported in the developed packages in section 2. Section 3 described the structure of the frameworks while section 4 presents the benchmark datasets used to validate the implemented similarity algorithms. Section 5 illustrates the evaluation procedure and the experimental results discussion.

2. Semantic similarity measures

A review of existing similarity measures that rely on the structure and content of English WordNet is presented in this section with a brief description about the Arabic knowledge source, AWN.

2.1. Arabic WordNet

A variety of algorithms adapted to Arabic in this paper use AWN to define the similarity, the only freely available lexical network of Arabic words. The AWN creation methodology was on the basis of the design and contents of the WordNet created for English (PWN). The AWN is organized into four principal structures which include item, word, form, and link. The item structure, also known as a conceptual entity contains synonym sets or synsets where each represents a lexical concept, synset-id, instances, and ontology class. The word structure represents a word sense which contains word form and word-id that used to associate a lemma with an item. The lexical information such as broken plural form and word's root were saved in the form structure. Finally, the link entity connects two items in relation such as equivalent, has-hyponym, related to, etc. The synsets of the AWN have been mapped to Suggested Upper Merged Ontology which is a language-independent ontology. This ontology classified the world into upper-level concepts (general concepts) [10]. These concepts were mapped to the more specific AWN synsets using three relations: subsumption, equivalent, and instance links.
The AWN version used in this paper comprises 11,270 synsts containing a total of about 23,496 unique Arabic words. These synsets cover different parts of speech: nouns, verbs, adjectives, and adverbs [11]. The nouns

and verbs are in separate subsumption (is-a) hierarchies. The noun hierarchy considers a rich hierarchy where the noun synsets were organized into nine taxonomies each has a top (root) known as a unique beginner. The maximum depth of this hierarchy is 15 nodes. Most of the algorithms that will be discussed in this paper exploited the noun hierarchy to identify the similarity.
In discussing Arabic wordnet, the following terms and definitions will be used:

- The shortest path length in AWN between synset c_i and synset c_j is denoted by $SPlen\left(c_i, c_j\right)$ and can be measured in edge counting or node counting.
- The depth of a node represents the path length between that node and the root (unique beginner) in which the node is located.
$$dep(c_i) = \text{len}(root, c_i)$$
- The Least Common Subsumer of c_1 and c_2 is denoted by $LCS(c_1, c_2)$, it is the meeting point that subsumes the two synsets. All the noun taxonomies will be joined into one taxonomy by adding a unique root node (virtual node) to ensure the existence of the LCS between any two nodes. This behavior will be turned on only with the similarity algorithm proposed by Leacock and Chodorow [12] while it will be turned off with others.
- The semantic similarity between two concepts is denoted by $Sim(c_1, c_2)$for applications that have sense tagged data, while the similarity between two words is denoted by $Sim(w_1, w_2)$ and can be calculated using (1).

$$Sim(w_1, w_2) = \max\left(Sim(c_1, c_2)\right), \quad c_1 \epsilon\ s(w_1)\ and\ c_2 \epsilon\ s(w_2) \tag{1}$$

Where, $s(w_i)$ represents the set of concepts in the noun hierarchy and this set consider the senses of word w_i. Following ***Resnik 1995*** [13], the similarity between two words is equal to the maximum senses similarity obtained from the pair of senses of the two words.

2.2. Path length similarity measures

In WordNet, a simple method to identify the similarity of concepts is to calculate the shortest path length connecting them in the is-a hierarchy Rada et al. 1989 [14]. The longer the path (distance) between concepts, the less similar the compared concepts are. Rada used the edge counting to calculate the length of the shortest path between the compared concepts. Given two concepts c1 and c2 with the consideration of all the possible paths between them, the semantic distance is defined as follows:

$$dis_{Rad}(c_1,\ c_2) = SPlen(c_1, c_2) \tag{2}$$

The semantic similarity between the compared concepts is computed using (3) while the semantic similarity between two words is calculated using (4):

$$Sim_{Rad}(c_1,\ c_2) = {}^{1}\!/_{dis_{Rad}(c_1,\ c_2)} \tag{3}$$

$$Sim_{Rad}(w_1, w_2) = \max\left(Sim_{Rad}(c_1, c_2)\right), \quad c_1 \epsilon\ s(w_1)\ and\ c_2 \epsilon\ s(w_2) \tag{4}$$

Wu and Palmer [15] proposed a measurement for identifying the semantic similarity by exploiting the path length and depth of the compared concepts in a taxonomy. The computation process of the path length and depth was undertaken by finding LCS that subsumes the compared concepts. The node counting method was used to determine the path length between each of the compared concepts and LCS as well as the depth of LCS.

$$Sim_{Wup}(c_1, c_2) = \frac{2 * dep(LCS)}{\ln(c_1, LCS) + \ln(c_2, LCS) + 2 * dep(LCS)} \tag{5}$$

This measure was revised slightly by Resnik [16] using the edge counting method:

$$Sim_{Wup}(c_1, c_2) = \frac{2 * dep(LCS)}{dep(c_1) + dep(c_2)} \tag{6}$$

In our experiment, the original measure is implemented but the edge counting method is used to calculate the path length while the depth of LCS is calculated using the node counting method.

Leacock and Chodorow [12] similarity measure considers the shortest path length of the compared concepts and the maximum depth of the taxonomy. Using the edge counting method with the consideration of all the possible paths between the compared concepts, the semantic similarity is defined as follows

$$Sim_{Lch}(c_1, c_2) = -\log(\frac{SPln(c_1, c_2)}{2 . D}) \tag{7}$$

Where, D represents the maximum depth of the taxonomy. In our experiment and as described in section 2.1, the unique root node will be turned on with this measure so the maximum depth of the noun taxonomy in AWN is 15.

Zhong et al. [17] determined the similarity score by considering the depths of the compared concepts as well as the depth of the LCS. In their proposed measure, every concept in the hierarchy has a value known as 'milestone' which can be calculated using (8).

$$Milestone(c) = {}^{1/2}/_{K^{dep(c)}} \tag{8}$$

Where, k is a factor greater than 1. It represents a rate at which the value decreases along the hierarchy. Given two concepts c1 and c2, the similarity score is identified as follows

$$Dis_{Zhong}(c_1, c_2) = dis(c_1, LCS) + dis(c_2, LCS) \tag{9}$$

$$dis(c, LCS) = milestone(LCS) - milestone(c) \tag{10}$$

$$Sim_{Zhong}(c_1,\ c_2) = {}^{1}/_{dis_{Zhong}(c_1,\ c_2)} \tag{11}$$

This measure used the edge counting method.

Pekar and Staab [18] identified the similarity score by considering the shortest path length between two synsets. The similarity of the compared concepts is directly proportional to the number of edges between the LCS and the root. Given two concepts c1 and c2, the similarity score is identified as follows:

$$Sim_{Pks}(c_{1,}c_2) = SPln(LCS(c_1, c_2), c_1) + SPln(LCS(c_1, c_2), c_2) + SPln(LCS(c_1, c_2), root) \tag{12}$$

Faaza et al. [19] presented a nonlinear word similarity measure (AWSS) for Arabic language inspired by Li algorithm [20]. It is a structure-based measure which calculates the similarity based on the minimum path length and the depth of the meeting point LCS that subsumed the compared Arabic words. The length and depth were extracted from AWN. Edge counting method is used to calculate the shortest path length while the depth of LCS is determined using node counting method.

$$Sim(c_1, c_2) = e^{-\alpha l} * \tanh(\beta * d) \tag{13}$$

Where, α and β represents the weight factors of length and depth respectively.

2.3. Feature-based similarity measures

These measures utilize more semantic information than path length measures such as concept descriptions, taxonomic ancestors, etc. The similarity score is determined as a function of the compared concepts properties taking into consideration their common and non-common features. According to Tversky [21], common features of concepts lead to increase similarity score while non-common lead to reduce it.

Sánchez et al. [22] proposed a non-linear measure for estimating the similarity between the compared concepts as a function of semantic distance which assumes that concepts share many generalizations in common have less distance than concepts with a smaller amount. The similarity score of the compared concepts was calculated as a ratio between their distinctive taxonomic subsumers and the sum of the taxonomic subsumers of each of the compared concepts. The following formula was used with this measure.

$$dis(a,b) = log_2\left(1 + \frac{|A \setminus B| + |B \setminus A|}{|A \setminus B| + |B \setminus A| + |A \cap B|}\right) \quad (14)$$

Where A and B represent the set of taxonomical features (subsumers) of a and b.

2.4. Information content similarity measures

The measures in this category augment ontology's concepts with information content extracted from a corpus. The value of information content of each concept in the taxonomy is computed based on the occurrence of the concept in a corpus. The first measure of this category was proposed by Resnik 1995 [13]. The score of similarity was identified by calculating the information content of the meeting point that subsumed the two concepts in the taxonomy. The limitation of the Resink measure is that the concepts that share the same meeting point in the taxonomy are assigned the same similarity score.

Jiang and Conrath 1997 [23] and Lin 1998 [24] performed some modification to overcome the Resnik measure limitation by taking into account the information content of each of the compared concepts. Lin's measure identified the similarity as a ratio between the information shared by the compared concepts as Resnik and the sum of the information content of each concept.

Jiang and Conrath measure computed the distance between the compared concepts linearly where the sum of the information content of each concept subtracted from the information content of their meeting point as Resnik.

The similarity measures in this category require sense tagged corpus [25] for deriving the information content and this data currently is not available for Arabic. Consequently, these measures will not be included in the developed packages.

3. Methods

This paper presents two packages for measuring the semantic similarity between two Arabic words or concepts. These packages are designed to encourage and support Arabic researchers for developing and validating new methodologies along with comparing them with algorithms included in these packages. The developed packages contain Java / Perl modules were built with methods which take two concepts or words as input and return the similarity between them. The developed packages can be embedded within researchers Java/Perl programs as modules and calling their algorithms.

The developed packages implement seven semantic similarity algorithms proposed by Rada et al., Wu Palmar (*Wup*), Leacock and Chodorow (*Lch*), Zhong et al., Pekar and Staab (*Pks*), Faaza et al., and Sánchez et al. One of these algorithms (AWSS) was proposed for Arabic and the rest were proposed for English and adapted to Arabic using AWN. Six of the implemented algorithms are path-based category and only Sánchez algorithm belongs to feature-based category. The reason for selecting these algorithms to adapt them to Arabic is that these algorithms rely on English wordnet structure and content. The AWN creation methodology was on the basis of the design and contents of the English wordnet and thus makes adapting these algorithms to Arabic possible.

In our experiment, for the path-based algorithms, the edge counting method is used to calculate the shortest path length while the node counting method is used to determine the depth of LCS. In multiple inheritance case, all possible paths of the compared concepts are considered and the similarity is determined using the shortest path and/or the deepest LCS in the taxonomy. For feature-based algorithm and in multiple inheritance case, a set of all concepts subsumed each of the compared concepts that indicates its taxonomical features is determined.

Wsimilarity is the main class for all implemented modules in ***AWN-WordSimilarity*** package while ***Csimilarity*** is the main class for ***AWN-ConceptSimilarity*** package. ***AccessADB***, ***NormlizDM*** and ***LenDepFind*** are three modules providing all of the functionality needed to meet the supported algorithms requirements.

NormlizDM module contains methods for removing diacritics from AWN words ***ReDiac ()*** and performing normalization process ***ReMark ()*** to replace the letters with a dot, hamza (ء) or madda (~) in a given word with letters without in order to retrieve words from AWN, as will be described in section 5.1.

AccessADB module contains methods to access and retrieve data from AWN. ***GetIDSynset ()*** method is used to retrieve a list of synsetsid for a given search word based on its POS. The result will be a list of synsets-ids for the synsets the searching word is found in. ***GetHypsynSet ()*** method is created to return hypernyms for a given synsetid. ***GetRelatSyn ()*** method is created to return a list of synsets which is derivationally related to a given synsetid. ***GetHypAllWordConcept ()*** method is created to return the hypernyms for concepts containing a given words. ***GetSetFeature ()*** method is created to return a set of all concepts subsumed a given concept as its taxonomical features

LenDepFind module contains a method ***SPlen ()*** to find the shortest path length between two concepts in case of multiple inheritance, a method ***len ()*** to find the path length, and also a method ***depLCS ()*** to find the depth of LCS.

4. Dataset

The identification of the accuracy of a computational similarity measure can be done by comparing the measure performance with human perception [16] using a word benchmark dataset. First experiments for Arabic were conducted in 2012 [26] with participating of 82 native Arabic speakers. A list of 56 ordinary Arabic words was presented to 22 Arabic participants to generate a set of 70 word pairs. An experiment used a sample of 60 Arabic participants were carried out to judge the set of 70 word pairs on the basis of their similarity of meaning using a scale from 0 to 4. The similarity ratings collected for each of the 70 pairs of Arabic words were calculated as the average of the judgments given by 60 Arabic participants. The produced dataset is known as ANSS-70 and is employed to assess the accuracy of the similarity algorithms adapted to Arabic.
This dataset has been partitioned into two sub-datasets each of which consists of 35 Arabic word pairs [19]. One is known as the training dataset and the other known as the evaluation dataset. The training dataset is used for tuning algorithms parameters while the evaluation dataset is used to assess their accuracy. In our experiment, only AWSS measure requires determining its optimum parameters values.

5. Evaluation Procedure

5.1. Normalization process

In Arabic writing system, writing words require two types of symbols. These are letters and diacritics. In addition, some letters have similar shape and particular marks were added above or below these letters in order to discriminate them such as (madda (~), hamza (ء), and dot). The words were saved in AWN as lemmata with diacritics and marks which present an interesting challenge to the computation process of Arabic word similarity algorithm. The problem arises due to the contemporary writing system of Arabic where words are written without marks and diacritics which prevent retrieving words from AWN. Therefore, the decision was made to remove marks and diacritics from AWN lemmata where a normalization process was carried out to meet this requirement.

5.2. Experimental results and discussion

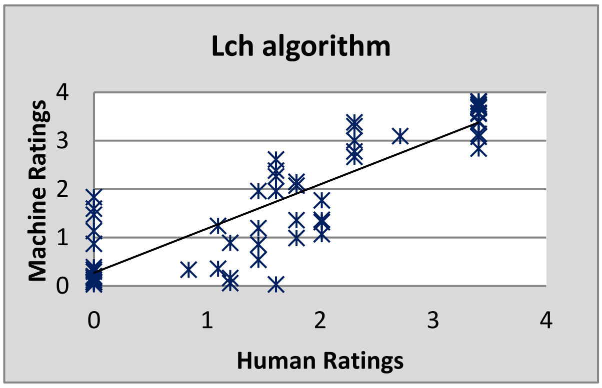
Two evaluation experiments are conducted in order to assess the accuracy of the similarity algorithms included in the developed packages. In first experiment, the accuracy of the adapted similarity algorithms are evaluated using ANSS-70 benchmark dataset described in section 4. The word pairs on ANSS-70 were run using the six similarity algorithms adapted to Arabic where each algorithm is given its author's name. Table 1 presents the results of this experiment where the correlations coefficients were computed between the human judgments

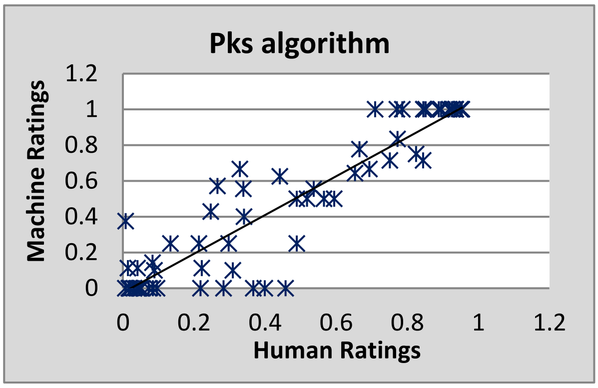
on ANSS-70 dataset and the ratings generated by each of the adapted similarity algorithms. Figure 1 shows the correlations coefficients of each of the adapted algorithms on ANSS-70 dataset.
As described in section 4, AWSS algorithm is the only one requires tuning its parameters. The role of these parameters was explored using the training dataset as stated in [19] while the evaluation dataset was employed to assess the AWSS measure accuracy. For the purpose of comparison, the word pairs on evaluation benchmark dataset were run using the adapted similarity algorithms. The results of the second experiment are presented in table 1 and the similarity ratings generated by the seven similarity algorithms with human ratings on evaluation dataset are presented in table 2.

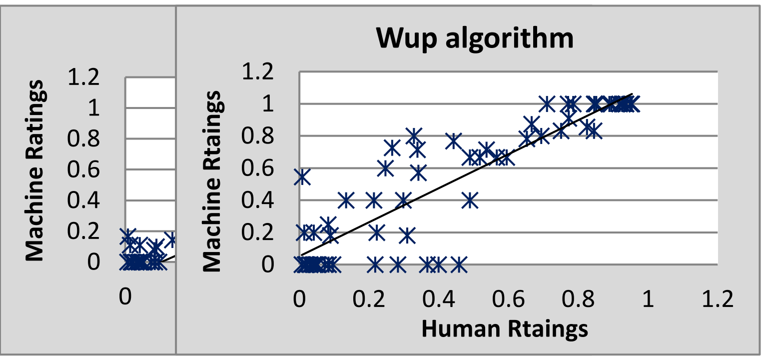
Table 1. correlation coefficient results

Algorithms	ANSS-70 Dataset	Evaluation Dataset
Average of the correlation of all participants	0.902	0.893
Lch algorithm	0.869	0.839
Rada algorithm	0.85	0.851
Pks algorithm	**0.912**	0.886
Wup algorithm	0.879	0.84
Zhong algorithm	0.891	0.887
Sanchez algorithm	0.902	**0.897**
AWSS algorithm	N/A	0.894

The possible performance's bounds expected from the algorithms of Arabic word similarity have been computed as the average of the correlations of all Arabic participants on the evaluation dataset and ANSS-70 dataset as shown in Table 1.
Pks algorithm achieved the best correlation coefficient at 0.912 on ANSS-70 dataset which exceeded the average of the Arabic participants' correlations at 0.902. The less correlation coefficient (at 0.85) was obtained by Rada algorithm on ANSS-70 dataset and its performance under the average of participants' correlations.
For the evaluation dataset, the best correlation achieved by Sanchez algorithm at 0.897 and the worst achieved by Lch algorithm at 0.839. Both Sanchez and AWSS algorithms achieved correlations exceed the average of the Arabic participants' correlations at 0.893 on evaluation dataset. In our experiment, Sanchez algorithm performed very well on the two benchmark datasets which achieved good correlations that equal or exceed the average of the correlations' participants, as shown in table 1.
This result indicates that it is possible to adapt these algorithms to Arabic and the developed packages are reliable to use and embed them with Arabic researchers' projects for improving or comparing their methodologies.







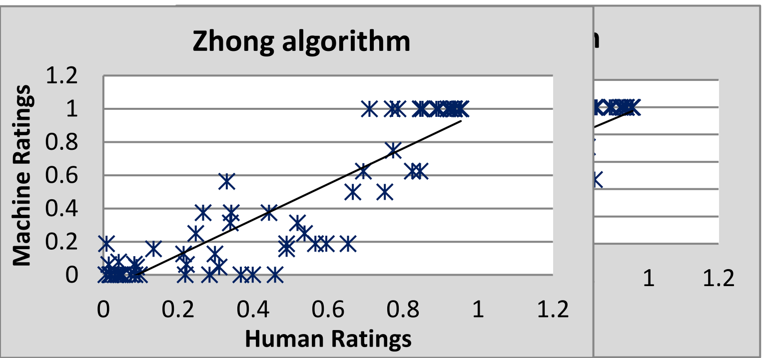


Figure 1. The correlations achieved by the adapted algorithms on ANSS-70 dataset.

Table 2. the similarity ratings from Human and the seven similarity algorithms on the evaluation dataset

No.	Word Pairs	Human Ratings	Lch	Rada	Pks	Wup	Sanchez	Zhong	AWSS	أزواج الكلمات
1	Coast Endorsement	0.01	0	0	0	0	0	0	0	ساحل تصديق
2	Noon String	0.01	1.61	0.17	0.38	0.55	0.3	0.19	0.27	ظهر خيط
3	Slave Vegetable	0.04	1.2	0.11	0.11	0.2	0.06	0.08	0.06	عبد خضار
4	Smile Village	0.05	0	0	0	0	0	0	0	ابتسامة/بسمة قرية
5	Hill Pigeon	0.08	0.84	0.08	0.14	0.25	0.11	0.06	0.06	تل حمامة
6	Glass Diamond	0.09	1.1	0.1	0.1	0.18	0.05	0.05	0.05	كأس الماس

No.	Word Pairs	Human Ratings	Lch	Rada	Pks	Wup	Sanchez	Zhong	AWSS	أزواج الكلمـــات
7	Cord Mountain	0.13	1.46	0.14	0.25	0.4	0.19	0.16	0.17	حبل جبل
8	Forest Shore	0.21	1.46	0.14	0.25	0.4	0.14	0.13	0.17	غابة شاطئ
9	sepulcher Sheikh	0.22	1.2	0.11	0.11	0.2	0.12	0.06	0.06	ضريح شيخ
10	Tool Pillow	0.25	1.79	0.2	0.43	0.6	0.35	0.25	0.32	أداة مخدة
11	Coast Mountain	0.27	2.01	0.25	0.57	0.73	0.49	0.38	0.45	ساحل جبل
12	Tool Tumbler	0.33	2.01	0.25	0.67	0.8	0.58	0.56	0.54	أداة قدح
13	Journey Shore	0.37	0	0	0	0	0	0	0	رحلة شاطئ
14	Coach Travel	0.40	0	0	0	0	0	0	0	حافلة سفر
15	Feast Fasting	0.49	1.46	0.14	0.25	0.4	0.19	0.16	0.17	عيد صيام
16	Coach Means	0.52	1.79	0.2	0.5	0.67	0.42	0.31	0.38	حافلة وسيلة
17	Girl Sister	0.60	1.61	0.17	0.5	0.67	0.46	0.19	0.37	فتاة اخت
18	Master Sheikh	0.67	2.3	0.33	0.78	0.88	0.58	0.5	0.67	سيد شيخ
19	Food Vegetable	0.69	2.3	0.33	0.67	0.8	0.58	0.63	0.53	طعام خضار
20	Slave Odalisque	0.71	3.4	1	1	1	1	1	0.93	عبد جارية
21	Run Walk	0.75	2.3	0.33	0.71	0.83	0.64	0.5	0.6	جري مشي
22	Cord String	0.77	2.71	0.5	0.83	0.91	0.78	0.75	0.7	حبل خيط
23	Forest Woodland	0.79	3.4	1	1	1	1	1	0.82	غابة أحراش
24	Cushion Pillow	0.85	3.4	1	1	1	1	1	0.82	مسند مخدة
25	Countryside Village	0.85	3.4	1	1	1	1	1	0.82	ريف قرية
26	Coast Shore	0.89	3.4	1	1	1	1	1	0.89	ساحل شاطئ
27	Tool Means	0.92	3.4	1	1	1	1	1	0.93	أداة وسيلة
28	Boy Lad	0.93	3.4	1	1	1	1	1	0.95	صبي فتى
29	Sepulcher Grave	0.94	3.4	1	1	1	1	1	0.82	ضريح قبر
30	Glass Tumbler	0.95	3.4	1	1	1	1	1	0.89	كأس قدح

6. Conclusion

Two packages known as AWN-WordSimilarity and AWN-ConceptSimilarity were presented for measuring the semantic similarity between two Arabic words or concepts. The developed packages contain Java / Perl modules for Linux/Windows OS users and can be embedded within researchers programs. Six English word similarity algorithms were successfully adapted to Arabic. Seven-word similarity algorithms were included in each package. The results of the developed packages validation process showed that these packages are reliable to use them and embed them with Arabic researchers' projects for improving or comparing their methodologies. These packages will be freely distributed through Source Forge which is an Open-Source development platform.

The frameworks described in this research represent the first step to providing a platform to be publicly available for developing and testing Arabic semantic similarity algorithms. Further research is required in future for including similarity algorithms from the information content-based category where ontology's concepts augmented with information content extracted from a corpus.

References

[1] M. R. Quillian. “Semantic memory,” in *Semantic inform. Processing*. M. Minsky. ED. Cambridge. MA: MIT Pres. 1968.

[2] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. WordNet::Similarity – measuring the relatedness of concepts. In *Proceedings of the Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 267–270, Boston, Massachusetts, 2004.

[3] G. A. Miller, "WordNet: A Lexical Database for English," Communications of the ACM, vol. 38, no. 1, pp.39–41, 1995.

[4] Z. Zhang, J. Otterbacher, and D. Radev. Learning cross-document structural relationships using boosting. *In Proceedings of the 12th International Conference on Information and Knowledge Management*, pages pp.124–130, 2003.

[5] D. McCarthy, R. Koeling, and J. Weeds. Ranking WordNet senses automatically. Technical Report CSRP 569, University of Sussex, January, 2004.

[6] W. H. Gomaa and A. A. Fahmy, "Automatic Scoring for Answers to Arabic Test Questions," Computer Speech & Language, vol. 28, no. 4, pp.833-857, 2014.

[7] Ted Pedersen, Duluth at semeval-2017 task 7: Puns upon a midnight dreary, lexical semantics for the weak and weary. *In Proceedings of the 11th International Workshop on Semantic Evaluation* (SemEval-2017), 2017.

[8] McInnes, B.T. and Pedersen, T. and Pakhomov, S.V. UMLS-Interface and UMLS-Similarity : Open Source Software for Measuring Paths and Semantic Similarity. *Proceedings of the American Medical Informatics Association (AMIA) Symposium*. Nonember, 2009, San Fransico, CA.

[9] S. Elkateb, W. Black, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease and C. Fellbaum, "Building a Wordnet for Arabic," In the fifth international conference on Language Resources and Evaluation (LREC), 2006.

[10] PEASE, A., NILES, I. & LI, J. The suggested upper merged ontology: A large ontology for the semantic web and its applications. Working notes of the AAAI-2002 workshop on ontologies and the semantic web, 2002.

[11] Alkhalifa, M. & rodríguez, H. 2010. Automatically Extending Named Entities coverage of Arabic WordNet using Wikipedia. International Journal on Information and Communication Technologies.

[12] Leacock and Chodorow: Claudia Leacock and Martin Chodorow. Combining local context and WordNet similarity for word sense identification, in "Christianne Fellbaum. WordNet: An Electronic Lexical Database. The MIT press, 1998".

[13] Resnik, P. Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the 14th International Joint Conference on Artificial Intelligence, 448–453, 1995.

[14] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and Application of a Metric on Semantic Nets. IEEE Transactions on Systems, Man, and Cybernetics, vol.19, no.1, pp. 17-30, January/February 1989.

[15] Z. Wu and M. Palmer. Verb Semantics and Lexical Selection. In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics (ACL'94), pp. 133-138, Las Cruces, New Mexico, 1994.

[16] P. Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity and Natural Language. Journal of Artificial Intelligence Research, vol. 11, pp. 95-130, 1999.

[17] Zhong, J. and Zhu, H. and Li, J. and Yu, Y. Conceptual graph matching for semantic search. *Proceedings of the 10th International Conference on Conceptual Structures*. pp. 92-106, 2002.

[18] Pekar, Viktor and Staab, Steffen. Taxonomy Learning: Factoring the Structure of a Taxonomy into a Semantic Classification Decision. Proceedings of the 19th International Conference on Computational Linguistics. COLING '02, 2002. Taipei, Taiwan, vol. 1, pp. 1-7. 2002.

[19] F. Almarsoomi, J. O'Shea, Z. Bandar and K. Crockett, "AWSS: an Algorithm for Measuring Arabic Word Semantic Similarity," In IEEE international conference on systems, man, and cybernetics, SMC, Manchester, United Kingdom, pp.504–509, 2013.

[20] Y. Li, Z. Bandar and D. Mclean, "An Approach for Measuring Semantic Similarity between Words using Multiple Information Sources," IEEE Transactions on Knowledge and Data Engineering, vol. 15, pp.871-882, 2003.

[21] Tversky, A. Features of Similarity. Psycological Review, vol. 84, pp. 327-352, 1977.

[22] Sánchez, D., Batet, M., Isern, D., Valls, A.: Ontology-based semantic similarity: A new feature-based approach. Expert Systems with Applications vol. 39, pp. 7718-7728, 2012.

[23] Jiang, J. J., & Conrath, D. W., Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In International Conference on Research in Computational Linguistics, ROCLING X pp. 19-33, 1997.

[24] Lin, D., An Information-Theoretic Definition of Similarity. In J. Shavlik (Ed.), Fifteenth International Conference on Machine Learning, ICML pp. 296-304, 1998. Madison, Wisconsin, USA: Morgan Kaufmann.

[25] Dhivya Chandrasekaran and Vijay Mago, Evolution of Semantic Similarity - A Survey. J. ACM 37, 4, Article 111, August 2020, 29 pages.

[26] F. Almarsoomi, J. O'Shea, Z. Bandar and K. Crockett, “Arabic Word Semantic Similarity,” in ICALLL (WASET), vol. 70, pp. 87–95, 2012.