Global Journal of Enterprise Information System

DOI: 10.18311/gjeis/2016/7655

A Semi-Supervised Graph-based Algorithm for Word Sense Disambiguation

Amita Jain[1*], Devendra Kumar Tayal[2] and Sonakshi Vij[3]

[1]Department of CS&E, AIACT&R, Geeta Colony, Delhi- 110031, India; amita_jain_17@yahoo.com

[2]Department of CS&E, IGDTUW, Kashmiri Gate, Delhi- 110006, India; dev_tayal2001@yahoo.com

[3]Department of CS&E, IGDTUW, Kashmiri Gate, Delhi- 110006, India; sonakshi.vij92@gmail.com

Abstract

Word sense disambiguation is an issue of computational linguistics that aims at extracting the most appropriate sense of a word in a given context. Till date, several unsupervised graph-based methods have been devised for achieving word sense disambiguation but the majority of these methods use the notion of using multiple ambiguous words in a text corpus to create a WordNet® graph which enforces the concept of "blind leading the blind". In this paper, a semi-supervised algorithm has been proposed and implemented that takes into consideration a clue-word for creating the desired WordNet® graph. The existing algorithms of word sense disambiguation consider all the graph connectivity measures to be equally significant but this is not the case. In this paper, a comparative study for all these graph connectivity measures is performed to discuss their connectivity aspects and priorities are assigned to them in order to generate an effective word sense disambiguation algorithm. The WordNet® graph is generated using python external libraries NetworkX and Matplotlib. The proposed algorithm's results are tested using SemCor database and it shows considerable improvement over the unsupervised graph-based method suggested by Navigli.

Keywords: Betweenness, Closeness, Degree, PageRank, Semi-Supervised Learning, WordNet®, Word Sense Disambiguation

(Date of Acceptance: 7-April-2016**; Plagiarism Check Date:** 12-April-2016**; Peer Reviewed by Three editors blindly:** 16-April-2016**; Reviewer's Comment send to author:** 23-May-2016**; Comment Incorporated and Revert by Author:** 26-June-2016**; Send for CRC:** 29-June-2016**)**

1. Introduction

One of the most prominent features of natural languages is the fact that they possess some kind of ambiguity[10]. This ambiguity, if not resolved could lead to a lot of miscommunications. Word sense disambiguation aims to solve this issue by using various learning techniques[11]. It refers to the task of finding the appropriate meaning of a word in the user's context. It is conceptually the same as word sense induction[12]. Various researchers have proposed several techniques to achieve it. The two major categories of learning include supervised and unsupervised learning. Supervised learning relies on the training datasets that are provided to the machine while in unsupervised learning the data is organized into different classes and no input training data set is needed[13]. Another technique which aims at extracting the advantages of both the supervised and unsupervised learning is semi-supervised learning[14]. It basically refers to the use of a large unlabeled dataset along with a given labeled data set so as to create some prediction rules that would give more precise results on the available data.

In general, the graph-based approach is a part of unsupervised learning but in the proposed algorithm, some useful input labels are considered for the priority values assigned to the centrality measures for the nodes of the graph which mark the presence of supervised learning. These values are assigned after performing a comparative study of these centrality measures. Hence, an effective semi-supervised algorithm is developed which can be applied in general to various circumstances which demand the knowledge about the correct and exact meaning of a word in a particular context. The local measures of centrality that this paper considers are Degree, Closeness, Betweenness, and PageRank. The semantic relations that are used for creating the WordNet® graph are hypernyms, hyponyms, meronyms, and holonyms.

2. Related Work

In 2002, S.Banerjee and T. Pedersen had presented an adaptive Lesk algorithm for achieving word sense disambiguation which utilized WordNet® as the sense inventory and tested the results against the data provided in Senseval-2[5]. In 2006, S.Patwardhan *et al.* had suggested the significance of using WordNet® based "context vectors" for determining the extent up to which two concepts are related[9]. In 2009, E. Agirre *et al.* had highlighted

*Author for correspondence

a way of "personalizing the PageRank method" in order to give more realistic results for word sense disambiguation on the given dataset[8].

In 2009, Navigli had conducted a well-elaborated survey on word sense disambiguation where the various approaches and techniques corresponding to the issue were explained in depth[1]. In 2010, Navigli *et al.* had presented the concept of BabelNet which is considered as a huge multilingual semantic network that combines the notions of WordNet® and Wikipedia[2]. In 2013, J.Wang has suggested the technique of using greedy max-cut for performing semi-supervised learning[7]. In 2014, Kingma *et al.* had suggested the significance of deep generative modeling using semi-supervised learning[6].

In 2015, Jain & Lobiyal had explained the concept of fuzzy Hindi WordNet® which is further used for performing word sense disambiguation using graph-based approach[3]. The major limitation of this algorithm was that it had considered all the semantic relations to be of equal significance. In 2010, Navigli & Lapata had provided a graph based approach for word sense disambiguation using various local and global measures of graph connectivity without considering the relevance of any semantic relation[4]. This paper extracts the advantages of all the relevant semantic relations and provides a semi-supervised WordNet® graph based algorithm for word sense disambiguation.

3. Graph Connectivity Measures

For measuring the significance of each node in a graph, various measures of graph connectivity are used which incorporates the concept of centrality in them[15]. These measures are discussed in detail in Table 1. If a vertex has higher centrality value then it is considered to be more significant than the other. The centrality measures used in our algorithm are for the weighted graphs[16]. These measures are:

3.1 Degree

This measure considers all the direct connections of a vertex. For weighted graphs the degree measure is calculated as the sum of all edge weights that are incident on it as is given by the following equation:

$$M_D = \sum_u w_{uv} \quad (1)$$

Where w_{uv} = weight of edge connecting node u and v
M_D = degree measure

3.2 Closeness

This measure considers the reciprocal of the total shortest distance from a given vertex to all other vertices. For weighted graphs this measure is given by the following equation:

$$M_C = \sum_{k=1\ tot} \frac{1}{w_k} \quad (2)$$

Where w_k= weight of edge connecting a node to k^{th} node
M_C = closeness measure

3.3 Betweenness

This measure is a representation of "how many pairs of nodes would have to go through a particular node in order to reach one another in the minimum number of hops"[17]. Hence, this measure has greater significance in terms of connectivity. For weighted graphs this measure is given by the following equation:

$$M_B = \sum_{s,t \in V; s \neq v \neq t} \frac{d_{st}(v)}{d_{st}} \quad (3)$$

Where $d_{st}(v)$ = weight/distance of edge connecting a node s and t
M_B= Betweenness measure

3.4 PageRank

This connectivity measure is somewhat different from other centrality measures. It is so because it acknowledges the fact that all connections of a node are not equal[18]. Some edges might be more significant than the others. For weighted graphs this measure is calculated using the following equation:

$$M_{PR} = (1-d) + d * \sum_{(V_a,V_b)\in E} \frac{w_{ba}}{\sum (V_c,V_b) \in E} PageRank(V_b) \quad (4)$$

Where w_{bc} = weight of edge connecting node b and a
w_{bc} = weight of edge connecting node b and c
M_{PR} = PageRank measure

In Table 1, the various aspects of the graph connectivity measures are discussed in detail.

Table 1 shows the comparison of various aspects of the Degree, Closeness, Betweenness and PageRank centrality measures. Betweenness and Closeness measures have better connectivity considerations as compared to the degree measure. This is so because degree centrality only considers the direct and immediate connections in a graph. Hence, it cannot broker between groups. On the other hand, PageRank acknowledges the fact that not all connections are equal.

4. Proposed Algorithm

This section discusses the proposed semi supervised graph based algorithm for disambiguating a word in English language.

Table 1. Comparison of various aspects of the measures of graph connectivity

Parameters	Degree	Closeness	Betweenness	PageRank
Basic Concept	A number of edges that terminate in a given vertex.	Defined as the reciprocal of the total shortest distance from a given vertex to all other vertices.	Defines how many pairs of vertices would have to go through a node in order to reach one another in the minimum number of hops.	Assigns relative scores to all vertices in the graph based on the Recursive principle.
What It Does	Gives a simple count of the number of connections a vertex has.	Tends to give high scores to vertices which are near the center of local clusters in an overall larger network.	Start by finding all the shortest paths between any two vertices in the graph and then count the number of these shortest paths that go through each vertex.	Acknowledges the fact that not all connections are equal.
Connectivity Aspects	It is unable to broker between groups.	Vertices which are highly connected to others within their own cluster will have a high closeness centrality.	If a vertex with high Betweenness is deleted from a network, the graph would fall apart into otherwise coherent clusters.	Based on the concept of Markov chain model.
Disadvantages	Only takes into account the immediate ties that a vertex has rather than indirect ties to all others.	Expresses only the average distance from each vertex to every other vertex in the graph.	High-Betweenness vertices often do not have the shortest average path to everyone else but they have the greatest number of shortest paths that necessarily have to go through them.	When a simple calculation is applied hundreds (or billions) of times over the results, it gets a bit complicated.

The algorithm initiates by taking the sample text as an input from the user and selecting the word which needs to be disambiguated. This word will be the target word. This sample text also contains the clue word that will be needed in the coming steps of the algorithm. This clue word will help in generation of a better WordNet® graph. In order to process the data further, text tokenization needs to be performed followed by part of speech tagging (POST). POST is majorly essential because it helps to associate a word with its corresponding "part of speech" and tells the user about how the further processing of this word will take place.

To create the WordNet® graph, assessment of the semantic relations and Synsets of the target word and clue word needs to be done. Hence, the tagged words are analyzed to generate a set of possible candidates for the clue word. A weighted WordNet® graph is drawn for all the clue words by considering the relevance of these semantic relations: Hypernym, Hyponym, Holonym, and Meronym. These relations are considered under the category of "parts of same speech". The graph is drawn using depth first search algorithm up to depth two i.e. we perform depth first search algorithm to include all edges that lead to a path between the target word and clue word using these Synsets and semantic relations up to depth two. Other depths are not feasible and relevant. If the depth is increased up to three then a WordNet® graph with thousands of nodes will be created which is irrelevant for calculations. The graph should ideally be dense but not too dense for calculations. The nodes in the graph should be relevant and less in number. The most feasible word is chosen according to the WordNet® graph so generated.

Once the clue word is selected, consider the WordNet® graph generated with its help to find the various centrality measures i.e. graph connectivity measures. These measures are then analyzed to find the most significant Synsets of the targeted word from the WordNet® graph. In the previous section various aspects of the centrality measures were discussed. On careful examination, the conclusion was drawn that degree centrality is the least important as far as word sense induction is being concerned. This is mainly due to the fact that it only considers the immediate connections of a given node. Also, Betweenness is the most significant measure. Closeness also plays an important role. Applications of PageRank lay midway. Hence, the proposed algorithm gives Betweenness the highest priority, followed by closeness, PageRank, and degree. The priority weight assignment is done as in Table 2. This marks the presence of semi supervised learning as labels are assigned to the concerned values in the form of priority weights.

Table 3 describes the proposed algorithm. All the centrality measures are calculated using equations (1), (2), (3) and (4). Now the significance score S_S is calculated. The Synset with the highest value of significance score will give the disambiguated sense for the target word.

The implementation and results obtained by this algorithm are discussed in detail in the next section.

Table 2. Priority weights of various centrality measures

Centrality Measure	Priority Weight Assigned
MD	1
MPR	2
MC	3
MB	4

5. Implementation and Results

This section demonstrates the execution of the proposed algorithm and discusses its implementation and results on SemCor database. The programming is done in python and the WordNet® graph is created using python external libraries NetworkX and Matplotlib. The text that is used for demonstration is "The employee was hired by the company". The word to be disambiguated is "company" i.e. the target word. Performing part of speech tagging on the sample text yields the following results:

Table 3. Proposed algorithm

Nomenclature :

a) Ambiguous word : X
b) Clue word: Y
c) Closeness measure : MC
d) Betweenness measure: MB
e) Degree measure : MD
f) PageRank measure : MPR
g) Significance score : SS
h) Number of elements in the set of possible clue words: N
i) Priority weight assigned to the centrality measure = w

1) START
2) User input : complete sentence
3) Choose X
4) Tag the words according to their "part of speech"
5) Generate the relevant set of possible clue words
6) If (N > 1)
   i. For each element: Specify Y
   ii. Draw the WordNet® graph as follows:
      a. Insert all the Synsets of X
      b. Extract the hypernyms, hyponyms, meronyms and holonyms for all the Synsets and insert them as nodes in the WordNet® graph
      c. Draw edges from these nodes to Y whenever there exists a path between them
      d. Add edge weights as follows:
         • Hypernyms → 1.0
         • Hyponyms → 0.8
         • Holonyms → 0.6
         • Meronyms → 0.4
   iii. If generated graph is in the form of a disconnected cluster / hugely dense graph: Discard Y and consider another candidate element to perform step a. to step d.
7) For the most relevant clue word calculate the following:
   a) MC
   b) MB
   c) MD
   d) MPR
8) Obtain the significant nodes/ Synsets from the calculated centrality values
9) Assign priority values to the centrality measures (Betweenness>Closeness>PageRank>Degree)
10) Calculate SS for each significant node as follows:

$$S_S = \sum_{i=1}^{4} wi * ci$$

11) The highest value (SS) node/Synset corresponds to the most relevant meaning
12) STOP

Tagged words= [('the', 'DT'), ('employee', 'NN'), ('was', 'VBD'), ('hired', 'VBN'), ('by', 'IN'), ('the', 'DT'), ('company', 'NN')]

Where: NN=Noun
DT=Determiner
VBD=Verb (Past Tense)
VBN=Verb (Past Principle)
IN=Subordinating conjunction or preposition

The set of possible clue words contain "employee" and "hired". A WordNet® graph is generated for both these clue words as shown in Figure 1 and 2. It can be seen that the WordNet® graph using "hired" as the clue word is not feasible for performing the required calculations as it is too dense. Hence, "employee" is used as the clue word to calculate the relevant measures of centrality.

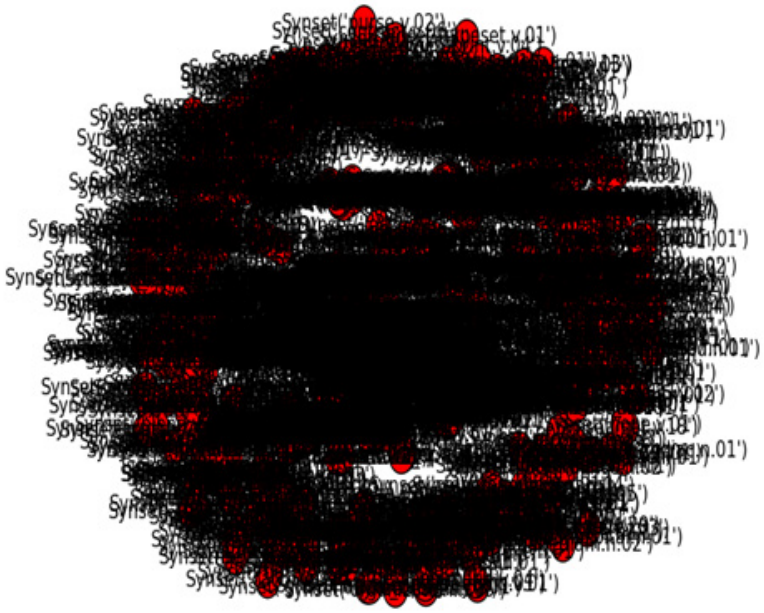

Figure 1. WordNet® graph considering "hired" as the clue word.

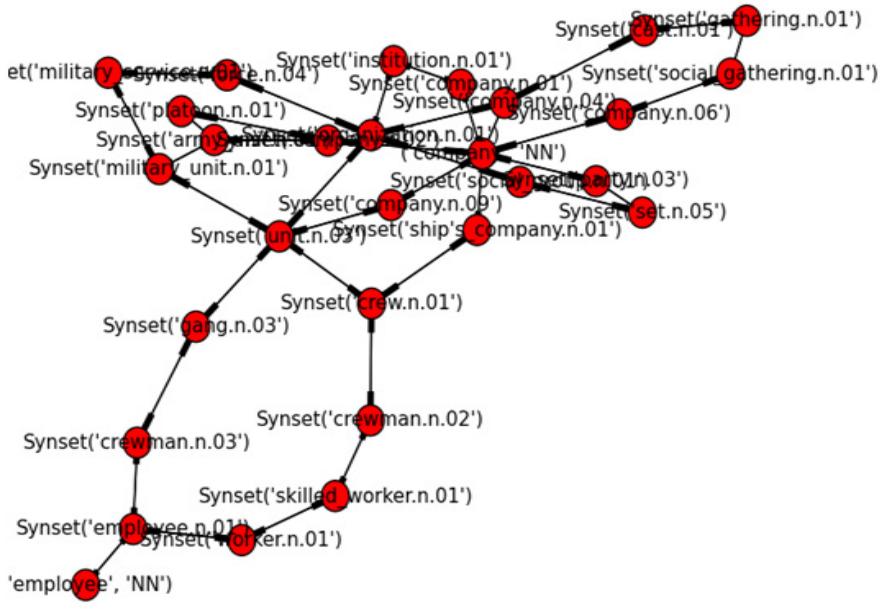

Figure 2. WordNet® graph considering "employee" as the clue word.

The results for all centrality measures for the WordNet® graph in Figure 2 are calculated and tabulated in Table 3. It can be analyzed from this table that the most significant nodes of the graph are "Company.n.02", "Company.n.04" and "Company.n.09" (marked in bold).

Table 3. Measures of centrality for the sample text

Nodes	M_D	M_C	M_B	M_{PR}
Worker.n.01	0.1429	0.2767	0.0198	0.0297
Crewman.n.03	0.1429	0.3743	0.1367	0.0289
Social_group.n.01	0.1429	0.3618	0.0397	0.0304
Force.n.04	0.1429	0.3938	0.0225	0.0286
Army_unit.n.01	0.2143	0.3733	0.0516	0.0385
Employee, NN	0.0714	0.2339	0.0	0.0224
Company.n.04	**0.2143**	**0.4451**	**0.1559**	**0.0347**
Employee.n.01	0.2143	0.3020	0.0970	0.0526
Organization.n.01	0.3571	0.4938	0.2958	0.0605
Crewman.n.02	0.1429	0.4000	0.1067	0.0289
Gang.n.03	0.1429	0.4288	0.1883	0.0234
Crew.n.01	0.2143	0.4628	0.1861	0.0305
Institution.n.01	0.1429	0.3972	0.0265	0.0241
Skilled_worker.n.01	0.1429	0.3185	0.0551	0.0335
Set.n.05	0.1429	0.3211	0.0123	0.0279
Company.n.01	0.1429	0.3753	0.0185	0.0278
Company, NN	0.5000	0.4930	0.3135	0.1022
Party.n.03	0.1429	0.3526	0.0388	0.0279
Company.n.09	**0.1429**	**0.4575**	**0.0727**	**0.0270**
Company.n.02	**0.2143**	**0.3794**	**0.0648**	**0.0353**
Military_service.n.01	0.2143	0.3840	0.0106	0.0260
Unit.n.03	0.3571	0.5385	0.4325	0.0588
Ship's_company.n.01	0.1429	0.4416	0.0791	0.0252
Company.n.06	0.1429	0.3500	0.0542	0.0285
Social_gathering.n.01	0.1429	0.2963	0.0040	0.0294
Cast.n.01	0.1429	0.3763	0.0780	0.0261
Platoon.n.01	0.1429	0.3382	0.0000	0.0219
Gathering.n.01	0.1429	0.2944	0.0159	0.0335
Military_unit.n.01	0.2143	0.4348	0.1243	0.0353

Now the priority weights are assigned to these measures of centrality as previously described in Table 2. The significance score (S_S) is then calculated as shown in Table 4. For the sake of simplicity, rounding off of the values up to two places of decimal is done.

From Table 4 it can be concluded that "Company.n.04" was the most significant node of the graph as it has the highest Significance Score (S_S) and hence it gives the most appropriate disambiguated sense for our target word. For obtaining the experimental results, the SemCor corpus is incorporated which is widely used for performing word sense disambiguation. Table 5 illustrates the results on SemCor database for all words and polysemous words in WordNet®. It shows that Betweenness outperforms the other measures of centrality for performing semi-supervised graph based word sense disambiguation.

The results obtained by using the proposed semi supervised algorithm are better than Navigli's unsupervised method which utilized various graph connectivity measures treating all of them to be equally relevant[4].

6. Conclusion and Future Scope

This paper presented a semi-supervised algorithm for word sense disambiguation algorithm. Semi-supervised learning provides two-sided advantages by exploiting the benefits of supervised and unsupervised learning. It also provides a way to select the most appropriate clue word in a given query that helps to initiate the disambiguation process. A weighted graph based approach for finding the intended meaning of a word in a particular context is used by using a priority based centrality measure calculation method that exploits the significance of various semantic relations. The results obtained for this algorithm are based on the

Table 4. Significance Score for various significant nodes

Node Details	M_D	M_C	M_B	M_{PR}	S_S
Company.n.02	0.21	0.38	0.06	0.04	1.67
Company.n.04	0.21	0.45	0.16	0.03	2.26
Company.n.09	0.14	0.46	0.07	0.03	1.86

Table 5. SemCor results

Measure	Navigli's method (All words)	Proposed method (All words)	Navigli's method (Polysemous words)	Proposed method (Polysemous words)
Degree	50.01	45.76	37.80	37.22
PageRank	49.76	49.88	37.49	37.55
Closeness	47.89	48.67	35.16	38.29
Betweenness	48.72	50.05	36.20	39.98

experiment on SemCor which shows that betweenness gave the best results, followed by closeness, PageRank, and degree. The results are better than the method proposed by Navigli which utilized unsupervised approach for word sense disambiguation. In future, this algorithm can further be extended by considering other semantic relations. Also, it may be extended to languages other than English.

7. References

1. Navigli R. Word sense disambiguation: A survey. ACM Computing Surveys (CSUR). 2009; 41(2):10.
2. Navigli R, Ponzetto SP. BabelNet: Building a very large multilingual semantic network. Proceedings of the 48th annual meeting of the association for computational linguistics; 2010. p. 216–25.
3. Jain A, Lobiyal DK. Fuzzy Hindi WordNet and Word Sense Disambiguation Using Fuzzy Graph Connectivity Measures. ACM Transactions on Asian and Low-Resource Language Information Processing. 2015; 15(2):8.
4. Navigli R, Lapata M. An experimental study of graph connectivity for unsupervised word sense disambiguation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010; 32(4):678–92.
5. Banerjee S, Pedersen T. An adapted Lesk algorithm for word sense disambiguation using WordNet®. Computational linguistics and intelligent text processing. Springer Berlin Heidelberg; 2002. p. 136–45.
6. Kingma DP, Mohamed S, Rezende DJ, Welling M. Semi-supervised learning with deep generative models. Advances in Neural Information Processing Systems. 2014. p. 3581–9.
7. Wang J, Jebara T, Chang SF. Semi-supervised learning using greedy max-cut. The Journal of Machine Learning Research. 2013; 14(1):771–800.
8. Agirre E, Soroa A. Personalizing PageRank for word sense disambiguation. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics; 2009 Mar. p. 33–41.
9. Patwardhan S, Pedersen T. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together. 2006; 1501:1–8.
10. Britton BK. Lexical ambiguity of words used in English text. Behavior research methods & Instrumentation. 1978; 10(1):1–7.
11. Lee YK, Ng HT. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. Proceedings of the ACL-02 conference on Empirical methods in natural language processing. 2002; 10:41–8.
12. Manandhar S, Klapaftis IP, Dligach D, Pradhan SS. SemEval-2010 task 14: Word sense induction & disambiguation. Proceedings of the 5th international workshop on semantic evaluation. 2010; 63–8.
13. Zhu X. Semi-supervised learning. Encyclopedia of machine learning. 2011. p. 892–7.
14. Borgatti, SP, Everett MG. A graph-theoretic perspective on centrality. Social networks. 2006; 28(4):466–84.
15. Newman ME. Analysis of weighted networks. Physical review E. 2004; 70(5):056131.
16. Brandes U. On variants of shortest-path betweenness centrality and their generic computation. Social Networks. 2008; 30(2):136–45.
17. Kurland O, Lee L. PageRank without hyperlinks: Structural reranking using links induced by language models. ACM Transactions on Information Systems (TOIS). 2010; 28(4):18.

Citation:

Amita Jain, Devendra Kumar Tayal and SonakshiVij
"A Semi-SupervisedGraph-based Algorithm for Word Sense Disambiguation",
Global Journal of Enterprise Information System. Volume-8, Issue-2, April-June, 2016. (http://informaticsjournals.com/index.php/gjeis)

Conflict of Interest:

Author of a paper had no conflict neither financially nor academically.