Providing FAQ Lists based on Ontology

Morteza Pourreza-Shahri
Graduate Student
Faculty of Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
pourreza@stu.um.ac.ir

Mohsen Kahani
Professor
Faculty of Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
kahani@um.ac.ir

Hamid-Reza Ekbia
Associate Professor
School of Informatics
Indiana University
Bloomington, IN, USA
hekbia@indiana.edu

Abstract— **Researchers have been fascinated in FAQ (Frequently Asked Questions) management systems in recent years. These systems have reduced the cost of supporting productions. The goal of this research is to implement a Persian FAQ generator in PC domain based on ontology. There are several steps for implementation. First, all slangs are converted into formal Persian phrases. Second, our proposed similarity method measures similarity between questions. After creating question similarity matrix, similar questions are gathered in the same clusters. FAQ lists are created based on clusters which have the most number of members. Evaluations indicates that our proposed similarity matrix generates the best results in comparison with Lin, MCS, and LSA matrices. Moreover, our proposed clustering method provides better results compared to hierarchical complete-linkage, hierarchical single-linkage, and Kmeans methods.**

Keywords; Frequently Asked Questions; Ontology; Clustering; FAQ list generation; Persian FAQ

I. INTRODUCTION

FAQ[1] management systems have attracted a great attention in last decades. Nowadays, customer satisfaction and product support is the highest priority for manufacturers. They spend a lot of time and money for training assistants to support their products. Therefore, the so-called Frequently Asked Questions systems were developed that stored users' asked questions. Every user could read questions and find the answer relating to his problem. However, that was a hard work reading all the questions without any search tools. Then, a search tool added to these lists but the problem was retrieving irrelevant answers. Most FAQ sites, however, provide no effective mechanisms to assist the user, who either has to pan through a long list of FAQs or has to rely on the use of the rudimentary keyword method to find relevant questions and answers [21]. Hence, the need for an automated FAQ management system increased significantly. Current systems use semantic methods to increase retrieving accuracy. These methods help users to enter their query in natural language and omit searching throughout all questions in the database.

Implementing a FAQ system requires two main tasks: 1) Creating FAQ lists 2) Retrieving related answers. We built domain ontology to help define domain vocabulary. In order to create FAQ lists, we introduced new clustering method based on Kmeans. We expanded Lee's sentence similarity measure [9] to work for our domain ontology.

The rest of the paper is organized as follows: Section 2 describes related works on FAQ lists. Section 3 explains the p. Section 4 reports the system evaluations, and Section 5 concludes the work. The Personal Computer (PC) domain is chosen as the target application of our FAQ system and will be used for any processing.

II. RELATED WORKS

There are three different techniques for semantic similarity measurement according to the linguistic level they can be applied to: sense, word, and text level.

Sense level measures for semantic similarity are mostly based on lexical resources. These measures have often viewed lexical resources as semantic and then used the structural properties of these networks in order to compute semantic similarity [14]. Wordnet [12] has an important role in semantic similarity between words. A comprehensive Wordnet-based measures is provided in [2]. In addition to Wordnet, other resources such as Wikipedia have played an important role in semantic measurement [3], [4].

The most attracting technique in the last decade is word-level similarity. There are two groups of word-level measures: distributional and lexical resource-based. A recent branch of distributional models uses neural networks to directly learn the context of a word [1]. In addition, lexical resource-based methods sometimes use words' closest senses to measure semantic similarity. Large collaborative datasets such as Wikipedia have been used in [19]. Lee's proposed two-phase algorithm [9] evaluates the semantic similarity for two or more sentences via a semantic vector space. The first phase built part-of-speech (POS) based subspaces by the raw data, and the latter carried out a cosine evaluation and adopted the WordNet ontology to construct the semantic vectors. Unlike other related researches that focused only on short sentences, Lee's algorithm is applicable to short (4–5 words), medium (8–12 words), and even long sentences (over 12 words) [9].

Third level is text-level techniques which can be grouped into two categories: (1) Those that view a text as a combination of words and calculate the similarity of two texts by

[1] Frequently Asked Questions

978-1-4673-8789-7/16/$31.00 ©2016 IEEE

aggregating the similarities of word pairs across the two texts, and (2) those that model a text as a whole and calculate the similarity of two texts by comparing its two models obtained. The methods in the first category usually calculate individual similarity values using large text corpora [8]. The second category computes the similarity of texts by comparing their corresponding vector [17].

III. PROPOSED METHOD

The proposed FAQ management system consists of two main phases. First, we calculate similarity score between questions and generate similarity matrix. Second, the proposed clustering method exploits FAQ lists based on similarity matrix. Fig. 1 shows the proposed system architecture. It is worth to explain some key points about the PC domain ontology, firstly.

A. Domain ontology

The concept of ontology in artificial intelligence refers to knowledge representation for domain-specific contents [5]. It is an important part of semantic based systems that supports knowledge sharing in developing intelligent systems. We outlined a procedure for developing ontology based on [20]. By following the steps, we developed a PC domain ontology using Protégé 2000 [13] as the fundamental part of our system. Fig. 2 shows the components of the ontology taxonomy. In this figure, PC concepts are classes and their parent–child relationships are donated by links, which indicate inheritance of features from parent classes to child classes.

B. Sentence similarity measurement

In this step, we calculate semantic similarity between questions. First of all, we need a measurement method. Although, Lee's algorithm works well for long sentences, it only breaks sentences into noun phrases and verb phrases and it cannot distinguish between the specific domain phrases and other noun phrases. So, in order to get better results Lee's algorithm needs to be adapted for our domain as follows.

We have two sentences A and B in the form of word sets:

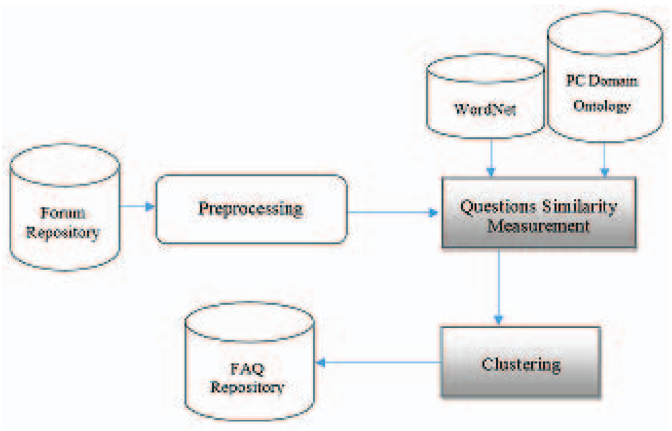

Figure 1. System Architecture

$$SEN_A = \{S\_V_A, S\_N_A, S\_C_A\} \tag{1}$$

$$SEN_B = \{S\_V_B, S\_N_B, S\_C_B\} \tag{2}$$

where S_V$_A$ is the verb set of sentence A, S_N$_A$ is the noun set of sentence A, and S_C$_A$ is our domain words set of sentence A. In the next step, we calculate verb vector (VV) of verbs corresponding to the base space (S_V$_A$ U S_V$_B$). We repeat this procedure for S_N and S_C as follows.

$$\left|NV_{SEN\_A}\right| = \left|NV_{SEN\_B}\right| = \left|S\_N_A \cup S\_N_B\right| \tag{3}$$

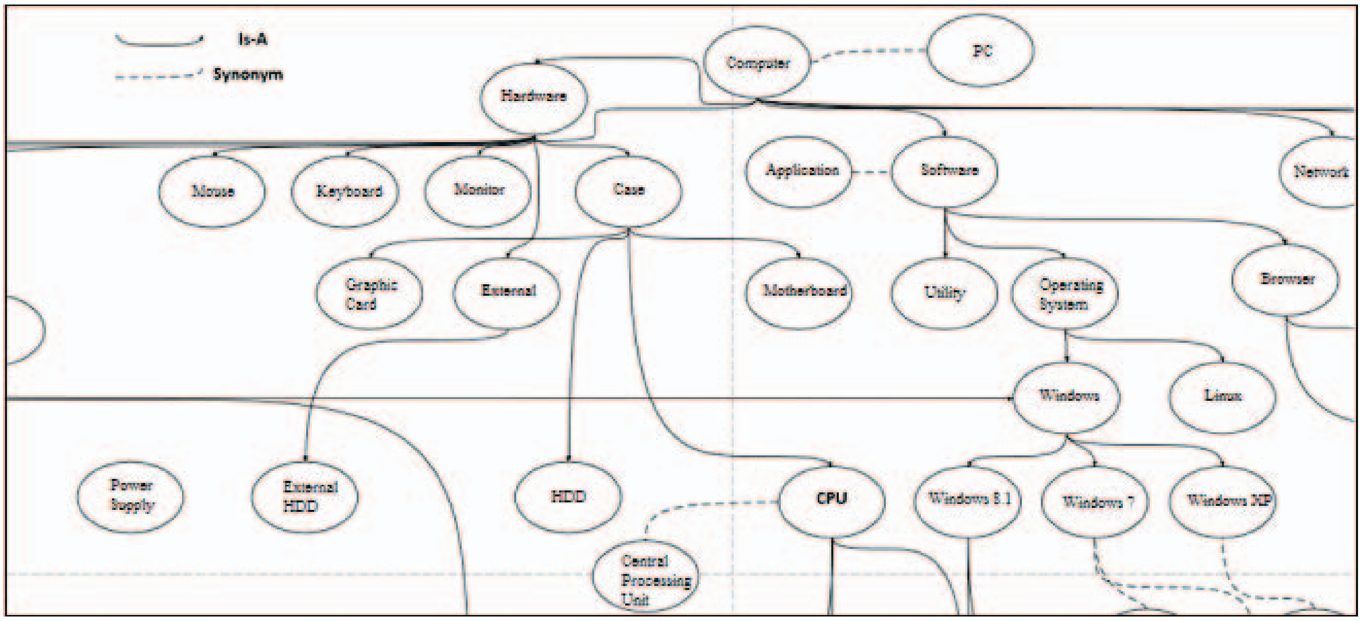

Figure 2. Components of PC domain ontology

$$\left|VV_{SEN\_A}\right| = \left|VV_{SEN\_B}\right| = \left|S\_V_A \cup S\_V_B\right| \tag{4}$$

$$\left|CV_{SEN\_A}\right| = \left|CV_{SEN\_B}\right| = \left|S\_C_A \cup S\_V_B\right| \tag{5}$$

where NV_{SEN_A} is the vector space of verbs in sentence A. VV_{SEN_A} and CV_{SEN_A} are the vector spaces of nouns and PC domain words in sentence A. The Noun Semantic Space (base space), and the Verb Semantic Space (base space) and Our PC domain Semantic Space are defined as the union of nouns in SEN_A and SEN_B, and the union of verbs in SEN_A and SEN_B and the union of domain words in SEN_A and SEN_B, respectively. The Wu & Palmer similarity measure [18] has become accepted as a standard for measuring similarity in lexical taxonomies. We adopt the Wu & Palmer similarity measurement to determine the similarity between two words. The formula is as follows:

$$\begin{aligned} &Similarity_{(WORD_A,WORD_B)} = 2\times DEPTH\left(H_l\right)\times \\ &(D_{Path\_Length}\left(WORD_A,H_l\right)+ \\ &\left(D_{Path\_Length}\left(WORD_B,H_l\right)+2\times DEPTH\left(H_l\right)\right)^{-1} \end{aligned} \tag{6}$$

where H_l is the depth of the lowest shared hypernym of $WORD_A$ and $WORD_B$. $DEPTH(H_l)$ is the level of H_l in the Wordnet semantic tree. $D_{Path\_Length}(WORD_A,H_l)$ is the semantic distance (number of hops) form H_l to $WORD_A$. $D_{Path\_Length}(WORD_B,H_l)$ is the semantic distance (number of hops) form H_l to $WORD_B$. Each word is compared to the base space to obtain the value of each field via formula (6).

In the next step, we calculate NV, VV, and CV for both sentences A and B.

$$\begin{aligned} &NV_{SEN\_A_i} = MAX_{k=1}^{|S\_N_A \cup S\_N_B|} \\ &\left(Similarity\left(Word_A, NOUN\_BASE_k\right)\right) \end{aligned} \tag{7}$$

$$\begin{aligned} &VV_{SEN\_A_i} = MAX_{k=1}^{|S\_V_A \cup S\_V_B|} \\ &\left(Similarity\left(Word_A, VERB\_BASE_k\right)\right) \end{aligned} \tag{8}$$

$$\begin{aligned} &CV_{SEN\_A_i} = MAX_{k=1}^{|S\_C_A \cup S\_C_B|} \\ &\left(Similarity\left(Word_A, COMPONENT\_BASE_k\right)\right) \end{aligned} \tag{9}$$

where NV_{SEN_Ai} denotes the value of NV of SEN_A in field i, and VV_{SEN_Ai} and CV_{SEN_Ai} denotes the values of VV of SEN_A in field i and CV of SEN_A in field i, respectively.

Then, we need to compute cosine angle of VV and NV and CV of the sentences, which are called Verb Cosine (VC) and Noun Cosine (NC) and Component Cosine (CC). We use traditional cosine measurement which is formulated as follows.

$$\begin{aligned} &NC_{A,B} = \left(\frac{\overrightarrow{NV}_{SEN\_A}.\overrightarrow{NV}_{SEN\_B}}{\left|\overrightarrow{NV}_{SEN\_A}\right|\times\left|\overrightarrow{NV}_{SEN\_B}\right|}\right)^2 = \\ &\left(\frac{\sum_{i=1}^{|S\_N_A \cup S\_N_B|} NV_{SEN\_A_i}\times NV_{SEN\_B_i}}{\sqrt{NV_{SEN\_A_i}{}^2}\times\sqrt{NV_{SEN\_B_i}{}^2}}\right)^2 \end{aligned} \tag{10}$$

$$\begin{aligned} &VC_{A,B} = \left(\frac{\overrightarrow{VV}_{SEN\_A}.\overrightarrow{VV}_{SEN\_B}}{\left|\overrightarrow{VV}_{SEN\_A}\right|\times\left|\overrightarrow{VV}_{SEN\_B}\right|}\right)^2 = \\ &\left(\frac{\sum_{i=1}^{|S\_V_A \cup S\_V_B|} VV_{SEN\_A_i}\times VV_{SEN\_B_i}}{\sqrt{VV_{SEN\_A_i}{}^2}\times\sqrt{VV_{SEN\_B_i}{}^2}}\right)^2 \end{aligned} \tag{11}$$

$$\begin{aligned} &CC_{A,B} = \left(\frac{\overrightarrow{CV}_{SEN\_A}.\overrightarrow{CV}_{SEN\_B}}{\left|\overrightarrow{CV}_{SEN\_A}\right|\times\left|\overrightarrow{CV}_{SEN\_B}\right|}\right)^2 = \\ &\left(\frac{\sum_{i=1}^{|S\_C_A \cup S\_C_B|} CV_{SEN\_A_i}\times CV_{SEN\_B_i}}{\sqrt{CV_{SEN\_A_i}{}^2}\times\sqrt{CV_{SEN\_B_i}{}^2}}\right)^2 \end{aligned} \tag{12}$$

Finally, we have three cosine measures (NC, VC and CC), and we need to combine them into an integrated score. The weights of NC, VC and CC are adjusted using coefficients α and β, which are determined by the users manually and via the experiment.

$$\begin{aligned} &Similarity_{A,B} = \propto \times NC_{A,B} + \\ &\beta\times VC_{A,B} + \left(1-\propto-\beta\right)\times CC_{A,B} \end{aligned} \tag{13}$$

There are two constraints for α and β:

$$\alpha,\beta \in [0,1]$$

$$\left(\propto + \beta\right) < 1$$

Now, we can create similarity matrices using semantic similarity scores of questions and then, cluster similar questions.

C. Exploiting FAQ list

The purpose of this step is exploiting FAQ list. Those questions which have been asked more frequent than others must be clustered together. For this purpose, we can use classification and clustering methods. Kmeans is a common method for clustering questions. But, Kmeans needs final number of clusters before the algorithm starts. We don't have the number of final clusters. If the initial number of clusters is too high, we may lose similar questions and if the number is too low, there are question which are not similar but in the same clusters. Therefore, we need to modify Kmeans method to work for our problem.

The modified Kmeans method is explained below.

1) *Step 1:* Calculate similarity score of each question with other questions and determine similar questions

2) *Step 2:* Create clusters for questions which have similar members more than a constant C. Center can be selected randomly.

3) *Step 3:* Sort clusters using the number of their members in descending.

4) *Step 4:* Omit members that belong to more than one cluster.

5) *Step 5:* Calculate similarity score of each question except cluster center with centers of other clusters.

6) *Step 6:* If similarity score is higher than constant S, add question to the cluster which have the higher value. Otherwise, create new cluster with that question as the center of cluster.

7) *Step 7:* Calculate Similarity score with other questions in the same cluster and select the member that has the highest value for similarity as new center.

8) *Step 8:* If at least one of centers has changed, then go to step 5. Otherwise, choose clusters which have members more than constant R as frequently asked questions.

Two questions are similar, if similarity value of them is higher than S. S is given to the algorithm manually. The higher threshold value for S, the more similar of the two questions would be. In this research, S = 0.7 was found to generate better results.

IV. SYSTEM EVALUATIONS

The proposed system was developed using Microsoft Visual Studio 2013 (C#.Net) and Matlab 8.1 (R2013a). There is no standard database for frequently asked questions in Persian. Therefore, the questions and answers in an online web forum called Gerdoo[2] were used. A set of 200 Q/As pairs was selected randomly for the evaluation. The evaluations are divided into two parts. First, evaluations on sentence similarity measure and Second, new clustering method evaluation.

A. Sentence Similarity Measurement Evaluation

For this purpose, as mentioned before, we used 200 questions and answers of Gerdoo web forum. We generated similarity matrix using our similarity method and this matrix was given to Kmeans, Hierarchical Single-Linkage, and Hierarchical Complete-Linkage methods. Meanwhile, we generated similarity matrices using Lin [10], MCS [11], and LSA[3] method and used those three clustering algorithms to cluster them.

Evaluation of the similarity methods was done by clustering quality methods such as Silhouette [15], Davies-Bouldin [6], Calinski-Harabasz [22], R-Squared [7], and Homogeneity-Separation [16]. We used CVAP 3.7[4] in Matlab to compute their values.

The outcomes of these algorithms are provided in tables I to III. In Silhouette method, higher values for S (Silhouette value) show better results. As can be seen, our similarity matrix generates better values in comparison with other three algorithms. In Davies-Bouldin method, lower values for DB (Davies-Bouldin value) show better results of algorithm and our similarity matrix generates the best results. In Calinski-Harabasz and R-Squared methods higher outputs are better that our matrix provides the best values. In Homogeneity-Separation method, lower value for Homogeneity and higher value for Separation show better clustering quality. Our similarity method generates the best values for both Ho (Homogeneity value) and Se (Separation value) in Complete-Linkage algorithm. In Kmeans and Single-Linkage algorithms, we have the best value for Homogeneity but Lin matrix generates better in Separation.

TABLE I. CLUSTERING QUALITY OF HIERARCHICAL COMPLETE-LINKAGE

	Input matrices for clustering			
	Our Matrix	***Lin***	***MCS***	***LSA***
S	0.3643	0.16842	0.16526	0.19404
DB	0.62724	1.1428	3.8571	1.1603
CH	9.73	3.1907	4.41	3.4482
R	0.9706	0.94131	0.8928	0.88311
Ho	0.9058	0.99774	0.93228	0.98122
Se	1.5385	1.3488	1.2663	1.3746

TABLE II. CLUSTERING QULALITY OF KMEANS

	Input matrices for clustering			
	Our Matrix	***Lin***	***MCS***	***LSA***
S	0.53788	0.48167	0.43397	0.50126
DB	0.22383	0.35241	0.3306	0.45585
CH	9.465	4.0128	8.8119	4.2765
R	0.97934	0.93487	0.98366	0.9419
Ho	1.1182	1.0803	1.1606	1.0846
Se	1.5018	1.3325	1.2501	1.3524

[2] www.gerdoo.net

[3] Latent Semantic Analysis

[4] www.cvap.com

TABLE III. CLUSTERING QUALITY OF HIERARCHICAL SINGLE-SLINKAGE

	Input matrices for clustering			
	Our Matrix	*Lin*	*MCS*	*LSA*
S	0.5959	0.43257	0.45346	0.36078
DB	0.27205	0.43696	0.50521	0.3704
CH	8.839	2.4486	3.8791	2.4503
R	0.96378	0.94118	0.92055	0.86437
Ho	1.568	1.0925	1.1497	1.124
Se	1.4878	1.3462	1.2507	1.3949

B. Evaluation of Clustering Method

In the second evaluation method, we needed to compare our clustering method with other common methods. For this purpose, we compared it with Hierarchical methods (both Complete-Linkage and Single Linkage) and Kmeans. First, our similarity matrix, Lin matrix, LSA matrix, and MCS matrix that we computed in the last section were fed as inputs to the clustering methods and then, we compare the clustering qualities using methods such as Silhouette, Davies-Bouldin, R-Squared, Calinski-Harabasz, and Homogeneity-Separation.

Table IV shows the results of the clustering algorithms using our similarity matrix as input. As can be seen from Table IV, our clustering method generates the best results for S, DB and CH (Calinski-Harabasz value). In R-Squared method, Kmeans has the best values and in Homogeneity-Separation, Complete-Linkage method is the best method.

Table V shows results from methods where input matrix is Lin matrix. Results are the same as our similarity matrix.

TABLE IV. CLUSTERING QUALITY USING OUR SIMILARITY MATRIX

	Our Method	Complete[a]	Single[b]	Kmeans
S	0.70481	0.3643	0.5959	0.26788
DB	0.53734	0.62724	0.27205	0.62383
CH	10.8264	9.73	8.839	6.465
R	0.96646	0.9706	0.97232	0.98422
Ho	1.6619	0.9058	1.568	1.3182
Se	1.4867	1.5385	1.4878	1.5018

a. Hierarchical (Complete-Linkage), b. Hierarchical (Single-Linkage)

TABLE V. CLUSTERING QUALITY USING LIN SIMILARITY MATRIX

	Our Method	Complete[a]	Single[b]	Kmeans
S	0.925	0.16842	0.43257	0.48167
DB	0. 3252	1.1428	0.43696	0.35241
CH	4.2774	3.1907	2.4486	4.0128
R	0.96621	0.94131	0.94118	0.98324
Ho	1.3354	0.99774	1.0925	1.0803
Se	1.3069	1.3488	1.3462	1.3325

a. Hierarchical (Complete-Linkage), b. Hierarchical (Single-Linkage)

TABLE VI. CLUSTERING QUALITY USING MCS SIMILARITY MATIRX

	Our Method	Complete[a]	Single[b]	Kmeans
S	0.47321	0.16526	0.45346	0.43397
DB	0.3245	3.8571	0.50521	0.3306
CH	9.8123	4.41	3.8791	8.8119
R	0.99354	0.8928	0.92055	0.98366
Ho	1.7319	0.93228	1.1497	1.1606
Se	1.2234	1.2663	1.2507	1.2501

a. Hierarchical (Complete-Linkage), b. Hierarchical (Single-Linkage)

TABLE VII. CLUSTERING QUALITY USING LSA SIMILARITY MATRIX

	Our Method	Complete[a]	Single[b]	Kmeans
S	0.52515	0.19404	0.36078	0.50126
DB	0. 3253	1.1603	0.3704	0.45585
CH	5.2029	3.4482	2.4503	4.2765
R	0.96635	0.88311	0.86437	0.9419
Ho	1.3515	0.98122	1.124	1.0846
Se	1.3331	1.3746	1.3949	1.3524

a. Hierarchical (Complete-Linkage), b. Hierarchical (Single-Linkage)

Table VI shows the results of evaluation algorithms using MCS matrix as input. As can be seen, our clustering method delivers the best values for S, DB, CH and R. But, Ho and Se values of Complete-Linkage are better than ours.

Table VII shows the results of clustering algorithms using LSA matrix. Our clustering method generates the best quality values for S, DB, CH and R. Single-Linkage method has the best value for Homogeneity, and Complete-Linkage method has the best value for Separation.

V. CONCLUSIONS

In this paper, a sentence similarity measurement and new clustering method for creating FAQ lists were presented. Our similarity method was given to common clustering methods and showed better results in comparison with other similarity matrices. Our proposed clustering method solved static number of initial clusters and provided good results compared to other common clustering methods.

REFERENCES

[1] M. Baroni, G. Dinu, and G. Kruszewski, “Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors”, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, pp. 238–247, 2014.

[2] A. Budanitsky and G. Hirst, “Evaluating WordNet-based Measures of Lexical Semantic Relatedness”, Computational Linguistics, vol. 32, no. 1, pp. 13-47, 2006.

[3] J. Camacho-Collados, M.T. Pilehvar, and R. Navigli, “NASARI: a novel approach to a semantically-aware representation of items”, Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, pp. 567–577, 2015.

[4] J. Camacho-Collados, M.T. Pilehvar, and R. Navigli, “A unified multilingual semantic representation of concepts”, Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, China, 2015.

[5] B. Chandrasekaran, J. Josephson, and V. Benjamins, “What are ontologies, and why do we need them?”, IEEE Intell. Syst., vol. 14, no. 1, pp. 20-26, 1999.

[6] D. Davies and D. Bouldin, “A Cluster Separation Measure”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, no. 2, pp. 224-227, 1979.

[7] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “On Clustering Validation Techniques”, Journal of Intelligent Information Systems, vol. 17, no. 23, pp. 107-145, 2001.

[8] A. L. Kashyap, L. Han, R. Yus, J. Sleeman, T. W. Satyapanich, S. R. Gandhi, and T. Finin, “Meerkat Mafia: multilingual and cross-level semantic textual similarity systems”, Proceedings of the 8th International Workshop on Semantic Evaluation, Dublin, Ireland, pp. 416–423, 2014.

[9] M. Lee, “A novel sentence similarity measure for semantic-based expert systems”, Expert Systems with Applications, vol. 38, no. 5, pp. 6392-6399, 2011.

[10] D. Lin, “An information-theoretic definition of similarity”, Proceedings of the 15th International Conference on Machine Learning, Madison,WI, pp. 296–304, July, 1998.

[11] R. Mihalcea, C. Corley, and C. Strapparava, “Corpus-based and Knowledge-based Measures of Text Semantic Similarity”, Proceedings of AAAI 2006, Boston, July, 2006.

[12] G.A. Miller, “Wordnet: a lexical database for english”, Communications of ACM, vol. 38, pp. 39–41, 1995.

[13] N. F. Noy and D. L. McGuinness, “Ontology development 101: A guide to creating your first ontology”, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI- 2001-0880, 2001.

[14] M. Pilehvar and R. Navigli, “From senses to texts: An all-in-one graph-based approach for measuring semantic similarity”, Artificial Intelligence, vol. 228, pp. 95-128, 2015.

[15] P. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis”, Journal of Computational and Applied Mathematics, vol. 20, pp. 53-65, 1987.

[16] R. Sharan, A. Maron-Katz, and R. Shamir, “CLICK and EXPANDER: a system for clustering and visualizing gene expression data”, Bioinformatics, vol. 19, no. 14, pp. 1787-1799, 2003.

[17] P. D. Turney, “Semantic composition and decomposition: from recognition to generation”, Tech. rep., National Research Council of Canada, 2014.

[18] Z. Wu and M. Palmer, “Verbs semantics and lexical selection”, Proceedings of the 32nd annual meeting on Association for Computational Linguistics, 1994.

[19] Z. Wu and C.L. Giles, “Sense-aware semantic analysis: a multi-prototype word representation model using Wikipedia”, Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, TX, USA, 2015.

[20] S. Y. Yang and C. S. Ho, “Ontology-supported user models for interface agents”, Proceedings of the 4th conference on artificial intelligence and applications, pp. 248–253, 1999.

[21] S. Yang, F. Chuang, and C. Ho, “Ontology-supported FAQ processing and ranking techniques”, J Intell Inf Syst, vol. 28, no. 3, pp. 233-251, 2007.

[22] Y. Zhao and G. Karypis, “Data Clustering in Life Sciences”, Molecular Biotechnology, vol. 31, no. 1, pp. 055-080, 2005.