Novel approach to Case Based Reasoning System by aggregating Semantic Similarity Measures using Fuzzy Aggregation for Case Retrieval

Riya A. Gandhi
Computer Department
L D College of Engineering
Ahmedabad, India.

VimalKumar B. Vaghela, PhD
Assistance Professor,
L D College of Engineering
Ahmedabad, India.

ABSTRACT
Natural language Search is used in Case Based Reasoning Systems for searching the solution to the novel problem. This paper presets the model of case based reasoning system that uses the semantic based case retrieval agent to compare two short texts. The proposed method include algorithms which calculate semantic similarity evaluated using different wordnet based semantic similarity measures and fuzzy aggregation. Based on the result, the proposed approach outperforms the results of previous approaches.

General Terms
Short texts, semantic similarity, Algorithms, RTE(Recognizing Textual entailment), Membership Functions, IF-THEN rules, Defuzzification.

Keywords
Case Based Reasoning Systems(CBR), Wordnet based semantic similarity measures, PATH, LCH, WUP, RES, JSN, LIN, Fuzzy Aggregation.

1. INTRODUCTION
Case Based Reasoning System uses Past experiences and past solutions to solve the new problem and to make the decision for the novel problem. CBR can be used in Problem Solving for design, planning, diagnosis and explanation[1]. The core of the CBR System is the case retrieval mechanism which uses similarity measures to match the current problem with the existing problem. Most of the measures calculate the similarity using string similarity but using string similarity it is difficult to find similarity when the meaning of two words is same but syntactically it is different. To overcome this limitation we use semantic similarity measures which can deal with the users through natural language. This paper presents the CBR model, in which we mainly designed the algorithms for semantic similarity which analyze user queries and calculates the similarity scores between two short texts using wordnet based semantic similarity measures. This study aims (1) to develop the CBR system which uses semantic similarity measures instead of string similarity for case retrieval (2) aggregate the scores of different semantic similarity measures with the help of fuzzy aggregation process (3) evaluate the performance of system with existing methods.

This paper organized as- Section 2 describe the background concepts, section 3 describe the proposed methodology and section 4 describe the experiment results and comparison with existing method.

2. BACKGROUND CONCEPTS
This section focus on introduction to CBR system and the concepts related to CBR system.

2.1 Case-Based Reasoning System
Case Based Reasoning is the process of solving users problem using past experiences, searching for the most related solution to the new problem and reusing that solution into new situations[1,3]. In CBR systems, we assume that the similar problems have similar solutions. It fetches most similar solution to our target problem, if it does not match perfectly then also we can get some basic idea or guidelines to solve our problem. The repository is used to store the solutions in the system which is called Case-base. The case-base contains set of problems, their solutions and information about how to solve the problem. CBR is the four step process[2].

Retrieve: A problem is given to the system, the system will retrieve the similar cases/solution from the case-base.

Reuse: The system will choose the possible solutions from the retrieved case, if retrieved solutions can not apply directly then they need to be adapted.

Revise: In this step existing solutions are modified to solve the target problem, it will continue to revise if necessary.

Retain: System will store the result in the case-base if the solution successfully used to solve the target problem.

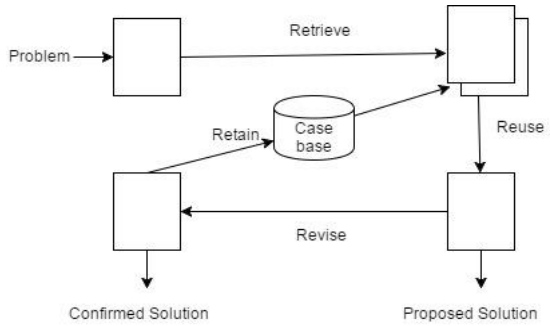

Fig.1 Traditional Case Based Reasoning System

2.2 Short Text and Semantic Similarity Measures
Short texts are basically defined as natural language search keywords or query which the user uses for search. It contains limited words. Most of the study shows that short text contains 1 to 8 words.

Semantic similarity measures are categorized into three types: corpus based, ontology based and hybrid[4]. The first method calculates the similarity from syntactic information and semantic information that they contain and this method is called STS(Semantic Text Similarity)[5]. The ontology based method is omiotis. It is an ontology based algorithm and

based on WordNet and WSD (Word-sense disambiguation). Omiotis uses various POS(part-of-speech) and semantic relations like synonymy, antonymy, hypernymy, etc. It extends Semantic Relatedness(SR) measure between the words[6]. SyMSS uses grammar parser to obtain the parse tree. It is the new method which considers the syntactic information and it uses this information in WSD(Word Sense Disambiguation) for reducing word matching and time complexity[7]. STATIS is the hybrid measure which combines WordNet based and corpus based word similarities.[8] Omiotis and SyMSS reduce the ambiguity between words using the syntactic information, POS and parse tree, respectively, to match words with the same syntactic role. The sentence semantic similarity measures are important in natural language research because of increasing applications in text-related research fields.

2.3 RTE(Recognizing Textual Entailment)

RTE is the task of recognizing that whether the meaning of one text can be inferred from another text or not. It is directional relation and generic task which captures the semantic relatedness across many natural language processing application. It is an asymmetric task for example, we can say that "the doctor is person." but "the person need not be a doctor." So it is asymmetric task but short text semantic similarity is symmetric task.

3. PROPOSED METHOD

This section consists of the design of case retrieval agent, the algorithms for finding similarity between two sentences and the fuzzy aggregation process.

3.1 Design of case retrieval mechanism

In the case based reasoning system, the most important part is the case retrieval mechanism which retrieves similar cases from the case base or repository. The requirement of user is given in natural language form. As shown in the fig. 2, First RTE is used to play the role of semantic similarity measure[5,7,8] and it checks whether the inputted requirement exist in the case base in the another form with the same meaning, if it is so then the similarity scores of that sentences are above the threshold so the solution is directly applied to the target problem. If similarity scores of the two sentences are not above the appropriate threshold, than Short Text Semantic Similarity measures are used to fetch the solutions which are meaning related with the user query. It also create new cases which will again be stored in case base for future use.

3.2 Semantic Similarity Algorithms used For Case Retrieval

For finding semantic similarity between two sentences, first we have to convert the natural language into semantic representation. We attach pos(parts of speech) to each word in the sentence. So the similarity between two short texts is called pos based short text semantic similarity which is based on wordnet based word measure[7]. There are six wordnet based word measure, which we will use to calculate the semantic similarity and at the end these six measures are aggregated using fuzzy aggregation to get the final score. The six measures are PATH[9], LCH, JCN[10], RES[11], LIN[12], WUP.

We convert the sentence into simplified POS tagset because the WordNet has only noun, verb, adverb and adjective. So first algorithm will convert the sentence into simplified POS sentence using StanFord Parser Based on Penn TreeBank which contain around 30 parts of speech(POS) tags[13]. The algorithm is as under.

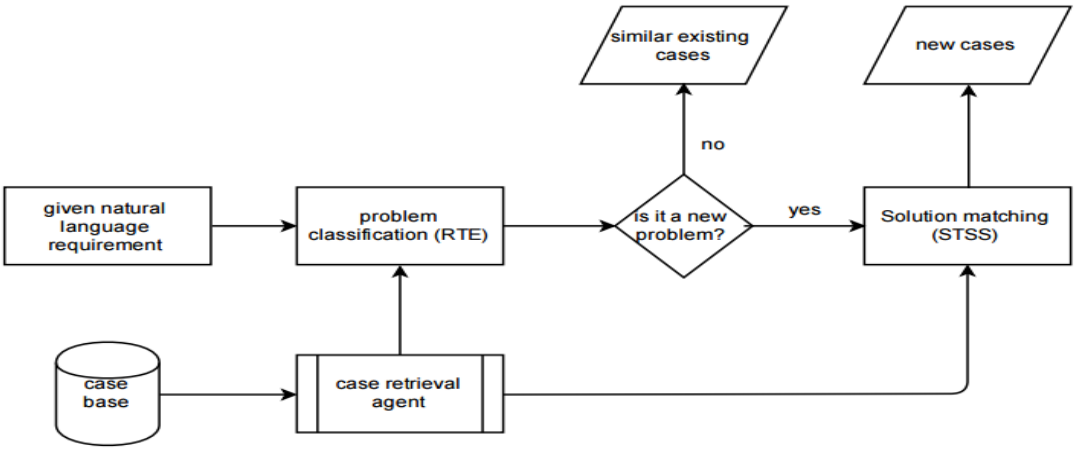

Fig.2. Case Retrieval Mechanism

Algorithm to generate simplified POS sentence[4]

INPUT: Sentence, Simplified POS table η

OUTPUT: Simplified POS Sentence

1. **Find named entity set from sentence and assign them unique Identifier ID#**
2. Apply stanford parser
3. For all tag_i ε Sentence
4. Do
5. Simplified POS_{sent} =LookupSimplifiedTag(η,Tag_i) END
6. RETURN Simplified POS_{sent}

Table 1. Simplified POS Tagset

Simplified POS	Penn Treebank POS
Noun (n)	NN, NNS, NNP, NNPS
Verb (v)	VB, VBD, VBG, VBN, VBP, VBZ
Adjective (a)	JJ, JJR, JJS
Adverb (r)	RB, RBR, RBS
Others (o)	CC, CD, DT, EX, FW, IN, LS, MD, PDT, POS, PRP, PRP$, RP, SYM, TO, UH, WDT, WP, WP$, WRB

The algorithm takes one sentence and the above table as input and convert the sentence into simplified POS tagset. The first step will find named entity From sentence and assign unique ID number to that entity[14]. For example if there are two short texts which are compared and one of the sentence contain the word "united states" and other contain the word "US", the meaning of both the words are same but representation is different so we will assign unique ID# number to that entity. After that the sentence applied to Stanford Parser and it will convert the sentence into simplified POS tagset.

Algorithm to calculate semantic similarity and aggregation of different similarity score

INPUT: $SimplifiedPOS_A$, $SimplifiedPOS_B$

OUTPUT: Similarity Score

1. ROW = MAX($SimplifiedPOS_A$, $SimplifiedPOS_B$)
2. COL = MIN($SimplifiedPOS_A$, $SimplifiedPOS_B$)
3. **n = [PATH, RES, JCN, WUP, LCH, LIN]**
4. $Length_A$ = Counting_words(S_A)
5. $Length_B$ = Counting_words(S_B)
6. FOR ALL c_x ε COL DO
7. FOR ALL r_y ε ROW DO
8. If $c_{x.}$ POS = $r_{y.}$ POS THEN
9. **FOR ALL n different measures**
10. SA[x]= max(SA[x], WordSimilarity(c_x .word , c_y .word,pos))
11. **END FOR**
12. END IF
13. END FOR
14. END FOR
15. **FOR ALL n different measure**
16. FOR 0 TO |COL|
17. MWS_{SUM} = MWS_{SUM} + SA[x]
18. END FOR
19. Normalization_Coefficient=($Length_A$+$Length_B$)/(2* $Length_A$*$Length_B$)
20. **END FOR**
21. **FUZZY AGGREGATION OF ALL NC for different n measure**
22. **RULE GENERATION**
23. **SimilarityScore =CoG**

The above algorithm takes two simplified POS sentences as input and gives the final similarity score based on fuzzy aggregation. First POS based coordinate matrix is formed based on the Word and POS. Assign the sentence having fewer words as row headers and other one as column headers of the matrix as shown in the fig.3. If two word has same POS then it is considered as word pair and the elements of the matrix computed using WordNet- based measure. For example S_AW_1–S_BW_1 and S_AW_3–S_BW_1 were the word pairs of Noun (Fig. 4).

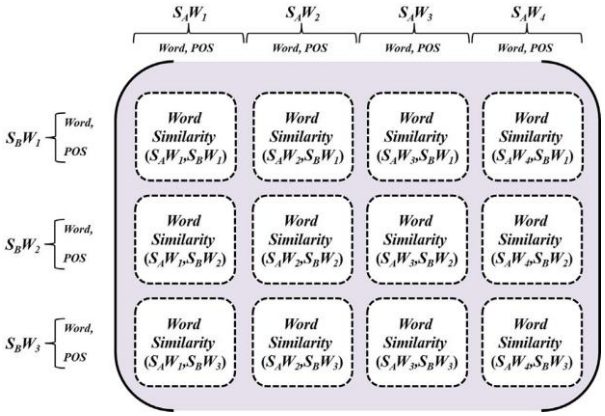

Fig.3.POS based coordinate matrix[4]

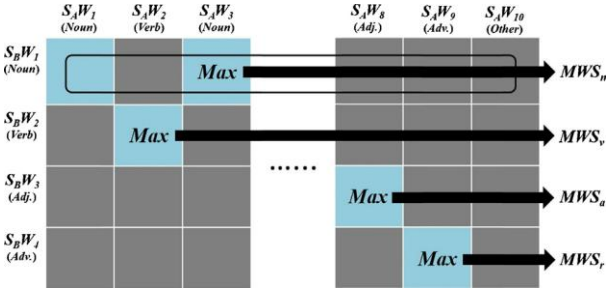

Fig.4.Semantic similarity optimization[4]

Word Similarity use PATH, LCH, JCN, RES, WUP and LIN measures to find the semantic similarity. The maximum similarity from each rows is extracted and we summed up these maximum word similarity of all rows to get maximum word similarity sum (MWS_{sum}). Then the similarity score is normalized using harmonic mean of the number of words in the sentence. The normalization coefficient is found for each of above six measures. The next step is to use fuzzy aggregation to aggregate all these six scores to get the result.

3.3 Fuzzy Aggregation Process

The different similarity scores are combined using fuzzy aggregation process. In Fuzzy aggregation process first we develop a fuzzy membership function for each measure to capture the importance of different semantic similarity measures, and then we use an operator for aggregation of multiple similarity measures. Fuzzy logic use linguistic values and expressions to describe numbers(similarity scores). The membership function states the membership of every element in the form of a numerical value between zero and one. We categorize these scores into three linguistic terms based on human expertise from literature survey which are bad, fair and excellent. Fuzzy aggregation process is designed by means of IF-THEN rules. These rules take these six variables as input.

Linguistic terms are bad (if two texts are not similar), fair (the two texts are moderately similar) and excellent (the two texts very much similar). Then, each linguistic term serve as an input for the rule engine which implements the aggregation. In a further step, based on the input, the rule engine triggers the rules that configure the resulting fuzzy set. Finally, the final aggregated score is retrieved by computing the CoG of the resulting fuzzy set[15]. For the defuzzification process, the method Center of Gravity (CoG) (fuzzy centroid method) is used to get final crisp value.

4. EXPERIMENTS AND RESULTS OF ALGORITHM

For this task, the dataset proposed by Li et al. [6] to enable comparison with other existing approaches. The dataset described by Li et al. [6] contains 65 sentence pairs created from 65 noun pairs, which are defined in the Collins Cobuild dictionary. Thirty sentence pairs were then selected by Li et al. for evaluation of this algorithm. This dataset contains the average similarity scores given by 32 human judges, and the human similarity scores are provided as the mean score for each sentence pair. To evaluate the performance of the method, pearson's correlation coefficient is used.

Following are some sample examples of sentences given in the dataset.

1. grin:implement

   "Grin is a broad smile."

   "An implement is a tool or other piece of equipment."

2. forest:woodland

   "A forest is a large area where trees grow close together."

   "Woodland is land with a lot of trees."

Table 2 shows the similarity scores of six different wordnet based word measures(PATH, JCN, LCH, WUP, RES, LIN) and the scores of fuzzy aggregation method with the human similarity scores. The table shows the Pearson's correlation coefficient value for each similarity measures and our method, which shows that by aggregating the value of different similarity measures we can get the highest value for pearson's correlation coefficient which is 0.85. The values of Pearson's correlation coefficient for different measures are PATH-0.83, LCH-0.82, WUP-0.79, RES-0.81, JCN-0.82, LIN-0.82.

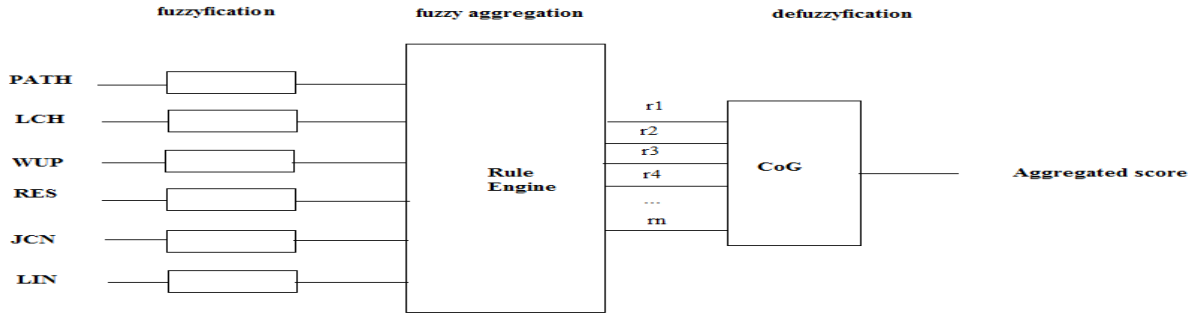

Fig.5. Fuzzy Aggregation of different similarity measures

Table 2. Result from existing system and fuzzy aggregation

No.	Human	Fuzzy Aggregation	PATH	LCH	WUP	RES	JCN	LIN
1	0.01	**0.183**	0.3	0.28	0.36	0.29	0.28	0.28
2	0.01	**0.39**	0.4	0.37	0.47	0.37	0.36	0.36
3	0.01	**0.298**	0.36	0.33	0.45	0.35	0.33	0.33
4	0.1	**0.5**	0.5	0.46	0.56	0.47	0.46	0.46
5	0.13	**0.185**	0.35	0.28	0.42	0.27	0.26	0.26
6	0.04	**0.176**	0.4	0.31	0.45	0.28	0.28	0.28
7	0.07	**0.319**	0.4	0.34	0.45	0.35	0.34	0.34
8	0.01	**0.39**	0.42	0.37	0.49	0.37	0.36	0.36
9	0.15	**0.5**	0.53	0.52	0.57	0.52	0.52	0.52
10	0.13	**0.38**	0.38	0.36	0.45	0.37	0.36	0.36
11	0.28	**0.41**	0.41	0.4	0.47	0.41	0.4	0.4
12	0.35	**0.48**	0.48	0.44	0.56	0.45	0.44	0.44
13	0.36	**0.5**	0.51	0.44	0.59	0.44	0.43	0.43
14	0.29	**0.5**	0.55	0.51	0.57	0.52	0.51	0.51
15	0.47	**0.44**	0.44	0.42	0.47	0.43	0.42	0.42
16	0.14	**0.44**	0.44	0.41	0.49	0.42	0.41	0.41
17	0.49	**0.5**	0.51	0.47	0.51	0.46	0.46	0.46
18	0.48	**0.5**	0.56	0.48	0.58	0.46	0.48	0.48
19	0.36	**0.5**	0.52	0.48	0.57	0.49	0.48	0.48
20	0.41	**0.44**	0.44	0.42	0.49	0.42	0.42	0.42
21	0.59	**0.55**	0.55	0.53	0.57	0.53	0.53	0.53
22	0.63	**0.48**	0.48	0.4	0.52	0.4	0.39	0.39
23	0.59	**0.57**	0.57	0.55	0.58	0.56	0.55	0.55
24	0.86	**0.94**	0.94	0.89	0.94	0.88	0.88	0.88
25	0.58	**0.58**	0.58	0.56	0.6	0.56	0.56	0.56
26	0.52	**0.5**	0.55	0.5	0.6	0.49	0.49	0.49
27	0.77	**0.59**	0.59	0.54	0.63	0.54	0.53	0.53
28	0.59	**0.56**	0.56	0.52	0.56	0.51	0.51	0.51
29	0.96	**0.95**	0.95	0.92	0.95	0.91	0.91	0.91
30	0.65	**0.633**	0.68	0.64	0.7	0.64	0.63	0.63
	Pearson's r	**0.85**	0.83	0.82	0.79	0.81	0.82	0.82

4.1 Comparison of Results

The below graphs shows the graphical representation of comparison. Fig.6 shows the comparison of Pearson's correlation coefficient and Fig.7 shows the comparison of similarity scores of different methods. It shows that fuzzy aggregation method gives the results which are very similar with human similarity so the pearson's correlation coefficient is higher than other method.

Thus we can achieve the best result when we aggregate different similarity scores using fuzzy aggregation process. By using Fuzzy Aggregation, we can add human intuition to this method because our goal is to get the similarity score which will nearly match with the human similarity.

5. CONCLUSION AND FUTURE WORK

Fuzzy aggregation is advantageous because in this atomic semantic similarity measure have to be aggregated without reflecting dissident values so it will remove the effect of poor similarity measure and we will get good similarity score. By identifying named entities , we come to know the actual semantic similarity of two short texts because it replaces the named entity by one ID so comparison becomes easy. Future work include the implementation of above algorithm with different datasets like Microsoft Paraphrase Corpus.

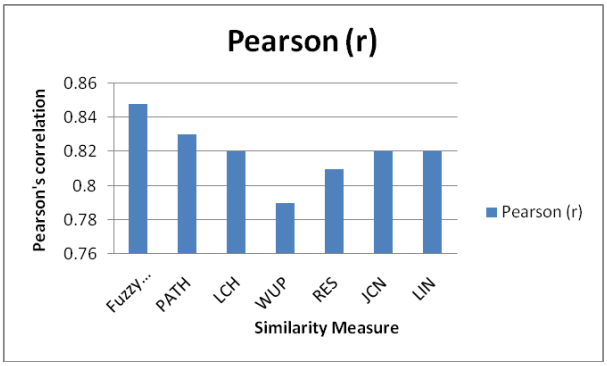

Fig.6. Comparison of Pearson's correlation coefficient of different similarity measure

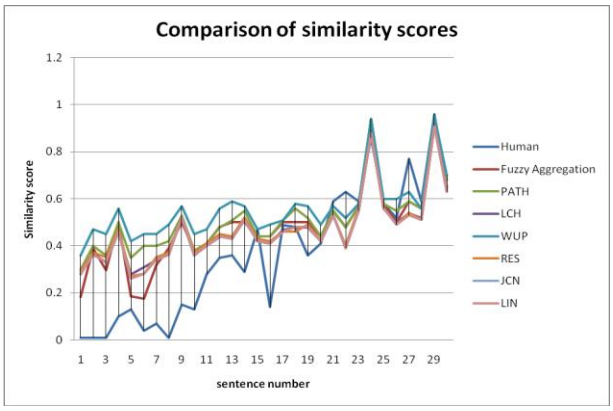

Fig.7. Comparison of different similarity measures with fuzzy aggregation

6. REFERENCES

[1] J.L. Kolodner, An introduction to case-based reasoning, Artif. Intell. Rev. 6 (1) (1992) 3–34.

[2] A. El-Fakdi, F. Gamero, J. Mele´ ndez, V. Auffret, P. Haigron, eXiTCDSS: a framework for a workflow-based CBR for interventional clinical decision support systems and its application to TAVI, Expert Syst. Appl. 41 (2) (2014) 284–294.

[3] A. Aamodt, E. Plaza, Case-based reasoning: foundational issues, methodological variations, and system approaches, AI Commun. 7 (1) (1994) 39–59.

[4] Jia Wei Chang , Ming Che Lee b, Tzone I Wang, "Integrating a semantic-based retrieval agent into case-based reasoning systems: A case study of an online bookstore," 2015 Elsevier,pp. 15–64.

[5] A. Islam, D. Inkpen, Semantic text similarity using corpus-based word similarity and string similarity, ACM Trans. Knowl. Discov. Data 2 (2) (2008) 1–25. [50] G. Tsatsaronis, I. Varlamis, M. Vazirgiannis, Text relatedness based on a word thesaurus, J. Artif. Intell. Res. 37 (1) (2010) 1–39..

[6] Y. Li, D. McLean, Z.A. Bandar, J.D. O'Shea, K. Crockett, Sentence similarity based on semantic nets and corpus statistics, IEEE Trans. Knowl. Data Eng. 18 (8) (2006) 1138–1150.

[7] J. Oliva, J.I. Serrano, M.D. del Castillo, A´. Iglesias, SyMSS: a syntax-based measure for short-text semantic similarity, Data Knowl. Eng. 70 (4) (2011) 390–405.

[8] G. Tsatsaronis, I. Varlamis, M. Vazirgiannis, Text relatedness based on a word thesaurus, J. Artif. Intell. Res. 37 (1) (2010) 1–39.

[9] R. Rada, H. Mili, E. Bicknell, M. Blettner, Development and application of a metric on semantic nets, IEEE Transactions on Systems, Man and Cybernetics 19 (1)(1987)17–30.

[10] J. Jiang, D. Conrath, Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, Proceedings on International Conference on Research in computational Linguistics, 1997, pp. 19–33.

[11] P. Resnik, Using Information Content to Evaluate Semantic Similarity in a Taxonomy, Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995, pp. 448–453.

[12] D. Lin, Using Syntactic Dependency as a Local Context to Resolve Word Sense Ambiguity, Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, 1997, pp. 64–71.

[13] M.P. Marcus, M.A. Marcinkiewicz, B. Santorini, Building a large annotated corpus of English: the Penn Treebank, Comput. Linguist. 19 (2) (1993) 313–330.

[14] Phuc H.Duong, Hien T. Nguyen, Ngoc-Tu Huynh, Measuring Similarity for Short Text on Social Media, Springer 2016.

[15] Jorge Martinez-Gil, CoTo:A novel Approach for fuzzy Aggregation of semantic Similarity Measures, ELSVIER 2016.

[16] J. Vanicˇek, I. Vrana , S. Aly, Fuzzy aggregation and averaging for group decision making:A generalization and survey, 2008 Elsevier, Knowledge-Based Systems 22 (2009) 79–84.

[17] Nasir Bedewi Siraj, Moataz Omar, Aminah Robinson Fayek, Combined Fuzzy Aggregation and Consensus Process for Multi-Criteria Group Decision Making Problems, IEEE 2016, 978-1-5090-4492.