Wireless Personal Communications (2020) 111:1271–1282
https://doi.org/10.1007/s11277-019-06913-x

A Machine Learning Approach for Automated Evaluation of Short Answers Using Text Similarity Based on WordNet Graphs

Sonakshi Vij[1] · Devendra Tayal[1] · Amita Jain[2]

Published online: 22 November 2019

Abstract
Answer sheet evaluation is a time-consuming task that requires lot of efforts by the teachers and hence there is a strong need of automation for the same. This paper proposes a machine learning based approach that relies on WordNet graphs for finding out the text similarity between the answer provided by the student and the ideal answer provided by the teacher to facilitate the automation of answer sheet evaluation. This work is the first attempt in the field of short answer-based evaluation using WordNet graphs. Here, a novel marking algorithm is provided which can incorporate semantic relations of the answer text into consideration. The results when tested on 400 answer sheets yield promising results as compared with the state-of-art.

Keywords Answer sheet evaluation · Graph theory · Machine learning · Text similarity · WordNet graph

1 Introduction

Answer sheet evaluations for the heart and soul of the examination system around the globe. Every now and then examinations are being carried out for various classes starting from nursery to higher education in the form of end-term examination, internal examination as well as weekly tests. This puts a lot of pressure on the evaluator to check and allot marks to the students in a given time frame. Since the evaluator is also involved in teaching, this leaves him/her very little time to completely dedicate themselves towards answer sheet evaluations. But this task is crucial and needs proper focus from the evaluator. Also, this needs to be an impartial task. It is common to see that sometimes due to the prejudices of the teacher; the marks of the student get affected.

✉ Devendra Tayal
dev_tayal2001@yahoo.com

1 Department of CSE, Indira Gandhi Delhi Technical University for Women, Delhi 110006, India

2 Department of CSE, Ambedkar Institute of Advanced Communication Technologies and Research, Delhi 110031, India

For instance, the teachers are slightly more inclined to giving good marks to the more obedient students. This is a psychological fact and can't be ignored.

In the recent years, technology has crept in the classrooms, making way for an efficient learning environment. From online lectures to e-classrooms, digitization has led to the evolution of a teacher and student friendly learning ambience. Automation of answer sheet evaluation has also been aimed in the past but high levels of accuracy have not been achieved. Earlier attempts in this field have either been not so accurate or they have been extensively time consuming. Through this paper, we aim at amalgamating the concepts of natural language processing such as context identification and text similarity into question answering, to facilitate the whole process of short answer-based script evaluation in an accurate and time efficient manner. Text similarity is a widely used technique for finding the amount of relatedness between two texts [1–9]. This concept has been studied by researchers worldwide for applications in various domains [10–15]. Text similarity approaches are being refined on a regular basis to provide optimum results for various applications [16–25].

Some of the latest research works in the field of short answer evaluation highlight that Fuzzy WordNet graphs play a significant role in the analysis. Vii et al. [26] depicts that the WordNet graph for the ideal answer can be generated and then it may be used to create a set of keywords that are essential for evaluating the students answer sheet. Since this paper uses WordNet as the sense repository hence it established context in a more elaborate manner. But this paper lacks the presentation of fruitful results as the testing is done only on a synthetic dataset. In order to add more relevance to such works, it becomes essential for us to include in this paper, a set of more elaborate results which are obtained after testing on a larger dataset.

In order to hand out meaningful explanations for sheet analysis, handwriting recognition can also be amalgamated in the process as depicted in [27]. Sijimol and Varghese [27] presents a model that can be used for acquiring a model that learns on the previous data based on the handwriting of an individual. But this is not a practical approach. Also, testing data is not sufficient for this analysis. Cosine sentence similarity was used in [27].

Van Hoecke [28] works on the algorithm that aims at utilizing sentence-based summarization techniques for performing grading of short answers of students. But this poses a limitation that sentence based ranking is not always accurate. So, the error due to faulty sentence ranking is relayed and carry forward in grading as well. Also, mostly sentence based ranking algorithms are based on machine translation and similarity scores which not very accurate. Hence, these types of approached are practically useful. Roy et al. [29] compares and contrasts the various existing techniques for performing short answer evaluation in terms of grading. This type of study is useful for us as it enables us to briefly outline the cons of the existing state-of-art techniques.

This paper focuses on proposing a novel method of automatically evaluating the answer sheets of students using a machine learning based approach. The technique adopted for the same is the generation of WordNet graphs. WordNet [5, 23] developed at Princeton university, is a computational lexicon consists of words and various relations between words. WordNet can be viewed as a graph where nodes represent words and edges represents relation between words. WordNet is widely being used in the literature for resolving several natural language processing tasks including word sense disambiguation, machine translation, information retrieval etc. [20, 21]. WordNet graphs play a significant role in information retrieval and hence they help in incorporating semantic significance and structural dependencies [22].

In this paper, WordNet is used for finding out the text similarity between the ideal answer provided by the teacher and the answer provided by the students in their answer sheet. The WordNet graphs are constructed to represent ideal answer and the answer for evaluation. Now the similarity between these two graphs is computed based on common nodes appearance. The marks for the answer to be evaluated are assigned in proportional to the similarity between these two graphs. The results for the proposed method are obtained on a dataset consisting of 400 students answer sheets. Answer sheets for evaluation are selected in a way to incorporate similarity and diversity in the data set.

The rest of the paper is framed as follows: Sect. 2 highlights the background study related to text similarity. Section 3 describes the proposed approach. Section 4 explains the results obtained. Section 5 concludes the work and states the relevant future scope.

2 Background Study

The main concept utilized in this paper for answer sheet evaluation is finding the text similarity between the ideal answer and the answer provided by the student. In order to study the latest recent trends in the field of text similarity, Web of Science (WoS) is taken as the data source. The below mentioned query was used for extracting the research papers pertaining to this field:

TI = (“Text Similarity”)

The research papers were obtained through the above-mentioned query from the year 1989–2017 [1–19]. The keywords occurring in these research papers were analyzed to visualize a keyword co-occurrence network as shown in Fig. 1. These keywords depict the various research topics associated with text similarity.

It can be observed from Fig. 1 that graph theory/technique, semantic dependencies and structural dependencies are closely associated to this field. Hence in this paper,

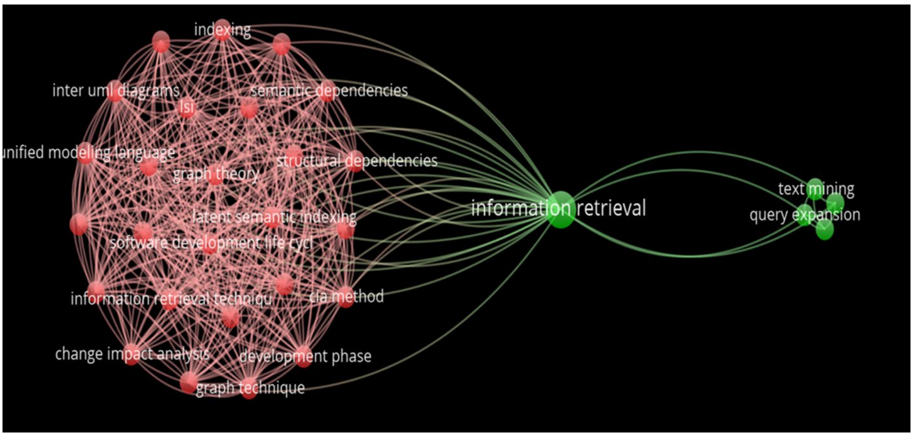


Fig. 1 Keyword co-occurrence network visualization for research papers in “text similarity”

a combination of these is taken to propose a novel method for calculating text similarity applied to answer sheet evaluation.

3 Proposed Approach

This section highlights the proposed approach adopted for evaluating the answer sheets of students in an automated manner. As concluded from the previous section, graph theory, semantic and structural dependencies play a significant role in text similarity calculation. Hence in this paper, a machine learning oriented WordNet graph-based method is proposed for answer sheet evaluation. WordNet is an online lexical dictionary type of database that consists of senses of a word according to its various part of speech tags. It consists of various semantic relationships intertwined to make a huge lexical database. WordNet graphs have been widely used in the literature for resolving lexical issues such as word sense disambiguation [20, 22]. The WordNet graph generated in this paper uses semantic relations hypernym, hyponym, meronym and holonym.

Siddiqi et al. [24] highlights that several types of short answer evaluations occur like the ones dealing with "True–False" type questions, fill in the blanks, sentence completion, "description required", "justification required", "example required" etc. the method proposed in this paper deals with short answer evaluation for the type of questions where a brief description is to be provided by the student with relevant short explanation if needed. The context can be well established for short answer evaluation using WordNet but for larger queries the context dissolves. For instance, it is difficult to automatically evaluate answers that have technical words in it since all of them are not available in WordNet. Other types of questions may be handled in the future. The method is explained as in Table 1.

To illustrate this, let us take the following text as the question to be evaluated:

Question: What is a car?

Table 1 Proposed method for automated answer sheet evaluation using WordNet graph-based text similarity

Nomenclature:

- N= Node set of common nodes between V_m and V_n
- **|N|**=Number of nodes in N i.e. its cardinality
- s = Total number of marks for a given question.
- m = Number of nodes in the WordNet graph of the ideal answer
- T_M = Total marks obtained by the student
- N_S = The node set for ideal answer from WordNet graph
- N_{S1}= The node set for candidate answer from WordNet graph
- C_J = Content words in the given text

i. **Step 1:** Input ideal answer by the evaluator/teacher and answer to be evaluated

ii. **Step 2:** Follow the steps to Generate WordNet graph for ideal answer:
- Obtain Synset of possible senses of C_J :
  $V \rightarrow \cup_{i=1 \text{ to } n} \text{Senses } (C_J)$
- For every node v ∈ V, perform DFS algorithm such as :
  - Use hypernym, hyponym, holonym and meronym to extract all nodes which lead to a path between v and v_k.
  - Add relevant nodes and corresponding edges to G

iii. **Step 3:** Obtain N_S

iv. **Step 4:** Following the similar procedure as in step 2 and step 3 for candidate answer and obtain N_{S1}

v. **Step 5:** Identify the common nodes in N_S and N_{S1} as:
$\boldsymbol{N} = N_S \cap N_S$
|N| // No. of common nodes

vi. **Step 6:** Marks evaluation is done as:
T_M= (**|N|** $*s)/m$
Where |N| is proportional to the marks assigned to the answer for evaluation

Answer: (Ideal, as provided by the teacher): Car is a vehicle with four wheels.

This answer text is treated as query Q_1. The implementation of the proposed method is carried out in python using Natural Language Tool Kit (NLTK) libraries.

Q_1 is tokenized and POS Tagging is done for the same. The result of the POS tagging set is as follows:

Tagged words set for Q_1 = [('car', 'NN'), ('is', 'VBZ'), ('a', 'DT'), ('vehicle', 'NN'), ('with', 'IN'), ('four', 'CD'), ('wheels', 'NNS')].

For the sake of simplicity, the content words chosen for generating the WordNet graph are ('car', 'NN'), ('vehicle', 'NN'), and ('wheels', 'NNS'). The semantic relations hypernym, hyponym, meronym and holonym are taken for this purpose. The WordNet graph is generated using depth first search algorithm [20]. It is shown as in Fig. 2 and has 49 nodes.

The node set of this graph (N_S) is as follows:

N_S = {"Synset('Wheeled_Vehicle.N.01')", "Synset('Car.N.01')", "Synset('Wheel.V.02')", "Synset('Wheel.N.01')", "Synset('Travel.V.01')", "Synset('Valve.N.03')", "Synset('Car_-Wheel.N.01')", "Synset('Handwheel.N.02')", "Synset('Rack.N.04')", "Synset-('Car.N.04')", "Synset('Cable_Car.N.01')", "Synset('Minivan.N.01')", "Synset-('Helm.N.01')", "Synset('Ride.V.02')", "Synset('Compartment.N.02')", "Synset-('Van.N.05')", "Synset('Vehicle.N.01')", "Synset('Steering_System.N.01')", "Synset-('Steering_Wheel.N.01')", "Synset('Wheel.V.03')", "Synset('Wagon_Wheel.-N.01')", "Synset('Lathe.N.01')", "('Vehicle', 'NN')", "Synset('Bicycle_Wheel.N.01')", "Synset('Instrumentality.N.03')", "Synset('Sprocket.N.02')", "Synset('Wheel.N.04')", "Synset('Conveyance.N.03')", "Synset('Cab.N.01')", "Synset('Bicycle.N.01')", "Synset-('Vehicle.N.03')", "Synset('Bicycle.V.01')", "Synset('Medium.N.01')", "Synset-('Passenger_Van.N.01')", "Synset('Vehicle.N.02')", "Synset('Motor_Vehicle.N.01')", "Synset('Roulette_Wheel.N.01')", "('Wheels', 'NNS')", "('Car', 'NN')", "Synset-('Wheel.N.03')", "Synset('Car.N.03')", "Synset('Fomite.N.01')", "Synset('Self-Propelled_

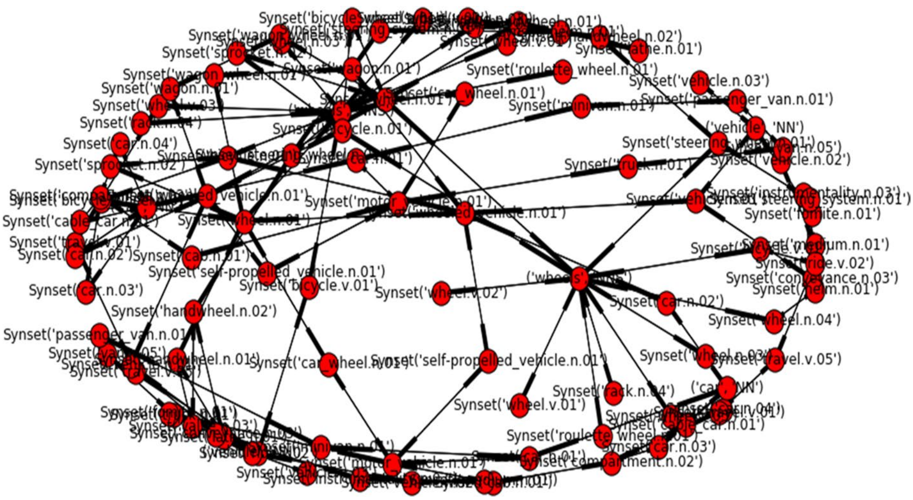


Fig. 2 WordNet graph for ideal answer

Vehicle.N.01')", "Synset('Wagon.N.01')", "Synset('Car.N.02')", "Synset('Handwheel.N.01')", "Synset('Travel.V.05')", "Synset('Wheel.V.01')", "Synset('Truck.N.01')"}

The answer sheets will be evaluated based on these nodes. There may exist, 2 basic types of answer sheets:

(a) When the answer written by the student matches logically with the ideal answer
(b) When the answer written by the student does not match with the ideal answer and is not relevant to the context either.

Case 1: When the student has written an accurate and logical answer according to the context

Let us suppose that the 1st candidate has put up the answer as:

Q_2: Car has wheels and an engine.

Now, in order to evaluate the 1st candidate answer sheet, Q_2 is tokenized and tagged as follows:

$$\text{Tagged words set for } Q_2 = [(\text{'car'}, \text{'NN'}), (\text{'has'}, \text{'VBZ'}), (\text{'wheels'}, \text{'NNS'}), (\text{'and'}, \text{'CC'}), (\text{'an'}, \text{'DT'}), (\text{'engine'}, \text{'NN'})]$$

where NN = Noun, VBZ = Verb, DT = Determiner, IN = Preposition, CD = Cardinal Digit

For the sake of simplicity, the content words chosen for generating the WordNet graph are ('car', 'NN'), ('wheels', 'NNS'), and ('engine', 'NN'). The WordNet graph is generated as shown as in Fig. 3. The total number of nodes in this WordNet graph is 53.

The node set of this graph (N_{S1}) is as follows:

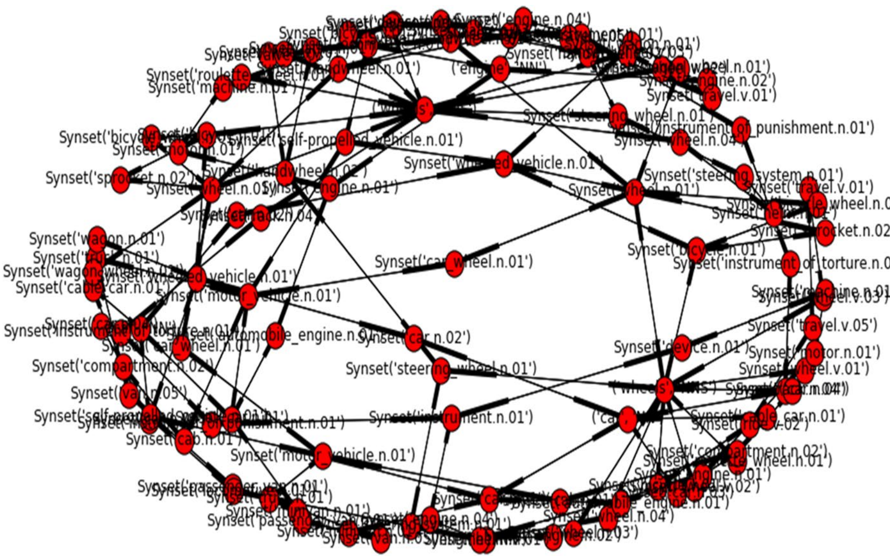


Fig. 3 WordNet graph for 1st candidate answer sheet

N_{S1} = {"Synset('Engine.N.02')", "Synset('Wheeled_Vehicle.N.01')", "Synset('Car.N.01')", "Synset('Wheel.V.02')", "Synset('Wheel.N.01')", "Synset('Travel.V.01')", "Synset-('Valve.N.03')", "Synset('Motor.N.01')", "Synset('Car_Wheel.N.01')", "Synset('Hand-wheel.N.02')", "Synset('Instrument_Of_Torture.N.01')", "('Engine', 'NN')", "Synset-('Rack.N.04')", "Synset('Engine.N.04')", "Synset('Car.N.04')", "Synset('Cable_Car.N.01')", "Synset('Minivan.N.01')", "Synset('Helm.N.01')", "Synset('Ride.V.02')", "Synset-('Compartment.N.02')", "Synset('Van.N.05')", "Synset('Automobile_Engine.N.01')", "Synset-('Steering_System.N.01')", "Synset('Steering_Wheel.N.01')", "Synset('Wheel.N.04')", "Synset('Locomotive.N.01')", "Synset('Instrument_Of_Punishment.N.01')", "Synset('Lathe.-N.01')", "Synset('Bicycle.V.01')", "Synset('Sprocket.N.02')", "Synset('Machine.N.01')", "Synset('Instrument.N.01')", "Synset('Bicycle_Wheel.N.01')", "('Wheels', 'NNS')", "Synset('Cab.N.01')", "Synset('Wagon_Wheel.N.01')", "Synset('Bicycle.N.01')", "Synset('Engine.N.01')", "Synset('Passenger_Van.N.01')", "Synset('Wheel.V.03')", "Synset('Motor_Vehicle.N.01')", "Synset('Roulette_Wheel.N.01')", "Synset-('Wheel.N.03')", "('Car', 'NN')", "Synset('Car.N.03')", "Synset('Travel.V.05')", "Synset-('Self-Propelled_Vehicle.N.01')", "Synset('Wagon.N.01')", "Synset('Device.N.01')", "Synset('Car.N.02')", "Synset('Handwheel.N.01')", "Synset('Wheel.V.01')", "Synset-('Truck.N.01')"}

Now, find out the nodes that match between N_{S1} and N_S and put them in N:

N = {"Synset('Wheeled_Vehicle.N.01')", "Synset('Car.N.01')", "Synset('Wheel.V.02')", "Synset('Wheel.N.01')", "Synset('Travel.V.01')", "Synset('Valve.N.03')", "Synset-('Car_Wheel.N.01')", "Synset('Handwheel.N.02')", "Synset('Rack.N.04')", "Synset-('Car.N.04')", "Synset('Cable_Car.N.01')", "Synset('Minivan.N.01')", "Synset('Helm.N.01')", "Synset('Ride.V.02')", "Synset('Compartment.N.02')", "Synset('Van.N.05')", "Synset-('Steering_System.N.01')", "Synset('Steering_Wheel.N.01')", "Synset('Wheel.N.04')", "Synset('Lathe.N.01')", "Synset('Bicycle.V.01')", "Synset('Sprocket.N.02')", "Synset-('Bicycle_Wheel.N.01')", "('Wheels', 'NNS')", "Synset('Cab.N.01')", "Synset-('Wagon_Wheel.N.01')", "Synset('Bicycle.N.01')", "Synset('Passenger_Van.N.01')", "Synset-('Wheel.V.03')", "Synset('Motor_Vehicle.N.01')", "Synset('Roulette_Wheel.N.01')", "Synset('Wheel.N.03')", "('Car', 'NN')", "Synset('Car.N.03')", "Synset('Self-Propelled_-Vehicle.N.01')", "Synset('Wagon.N.01')", "Synset('Car.N.02')", "Synset('Handwheel.N.01')", "Synset('Wheel.V.01')", "Synset('Truck.N.01')"}

It can be observed that N consists of 40 nodes (|N|) which means that out of 49 nodes in the ideal answer sheet graph, 40 matches with the 1st candidate answer sheet. This means that the answer is very relevant to the given context, and hence it can be marked for a 10-mark question as (40*10/49) = 8.1.

Case 2: When the answer written by the student does not match with the ideal answer and is not relevant to the context either.

Now the 2nd candidate answer sheet has put up the answer as:

Q_3: Car is used for transportation.

Now, in order to evaluate the 2nd candidate answer sheet, Q_3 is tokenized and tagged as follows:Tagged words set for Q_3 = [('car', 'NN'), ('is', 'VBZ'), ('used', 'VBN'), (‘for’, ‘IN’), (‘transportation’, ‘NN’)]where NN = Noun, VBZ = Verb, IN = Preposition, VBN = Verb (past participle)

For the sake of simplicity, the content words chosen for generating the WordNet graph are (‘car’, ‘NN’) and (‘transportation’, ‘NN’). The WordNet graph is generated as shown as in Fig. 4. The total number of nodes in this WordNet graph is 27.

The node set of this graph (N_{S2}) is as follows:

N_{S2}= {“Synset(‘Be.V.02’)”, “Synset(‘Exist.V.01’)”, “Synset(‘Equal.V.01’)”, “Synset-(‘Practice.V.04’)”, “Synset(‘Exploit.V.01’)”, “Synset(‘Be.V.10’)”, “Synset(‘Secondhand.S.02’)”, “Synset(‘Used.A.01’)”, “(‘Is’, ‘VBZ’)”, “Synset(‘Constitute.V.01’)”, “Synset(‘Be.V.12’)”, “Synset(‘Be.V.11’)”, “Synset(‘Exploited.S.02’)”, “Synset(‘Use.V.01’)”, “Synset(‘Use.V.02’)”, “Synset(‘Stay.V.01’)”, “Synset(‘Embody.V.02’)”, “Synset(‘Be.V.03’)”, “Synset-(‘Use.V.06’)”, “Synset(‘Cost.V.01’)”, “Synset(‘Take.V.02’)”, “Synset(‘Be.V.05’)”, “Synset(‘Use.V.03’)”, “Synset(‘Use.V.04’)”, “Synset(‘Be.V.01’)”, “Synset(‘Be.V.08’)”, “(‘Used’, ‘VBN’)”}

Now, find out the nodes that match between N_{S2} and N_S and put them in N:

$$N = \{\Phi\}//\text{NULL SET}$$

It can be observed that in this case, N doesn’t consist of any nodes which means that out of 49 nodes in the ideal answer sheet graph, none matches with the 2nd candidate answer sheet i.e. |N| = 0. This means that the answer is not relevant to the given context, and hence it would be marked zero. The results for the example taken for illustration are summarized as in Table 2.

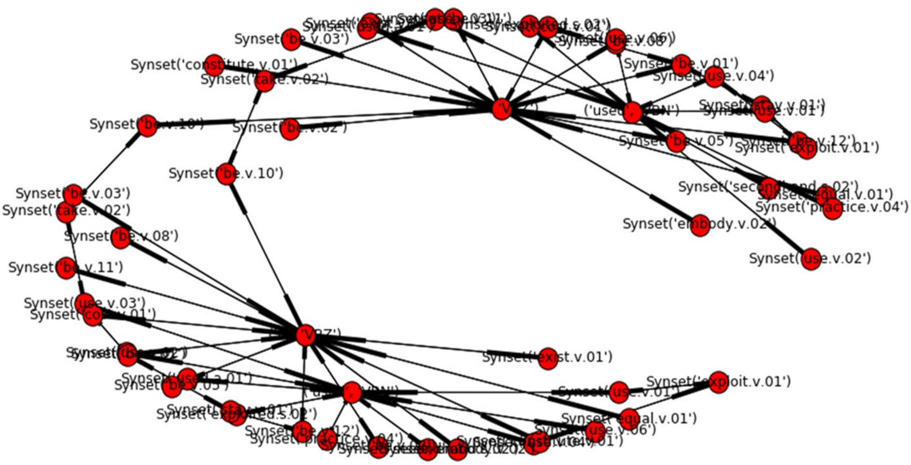
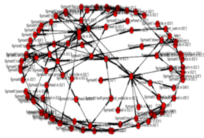
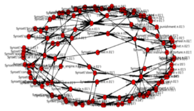
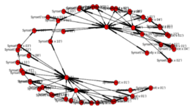


Fig. 4 WordNet graph for 2nd candidate answer sheet

Table 2 Results for the considered example

S. No.	Answer label	Answer	Wordnet graph	Number of common nodes	Marks allotted out of 10
1	Ideal	Car is a vehicle with four wheels		49	10
2	1st candidate	Car has wheels and an engine		40	8.1
3	2nd candidate	Car is used for transportation		0	0

4 Results and Evaluation

To test the effectiveness of this approach, a dataset was considered in which answer sheets of 400 students were collected. The answer sheets belong to the subject social studies. This was observed through experimentation that the proposed system does not apply well to technical subjects like computer science engineering. This is so because WordNet doesn't contain all the technical words and definitions. For the result evaluation, these 400 answer sheets were checked beforehand by the teachers. These sheets were scanned, and their text was converted into a machine-readable format using OCR (Optical Character Recognition). The answers in these sheets were analyzed according to the proposed method and were re-evaluated. The marks obtained by the proposed method and the actual marks were compared to calculate the Root Mean Square Error (*RMSE*) using Eq. 1.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (X_{obs,i} - X_{model,i})^2}{n}} \tag{1}$$

here $X_{obs,i}$ = Marks of the answer sheet as evaluated by the teacher, $X_{model,i}$ = Marks of the answer sheet as calculated by the proposed method, n = Number of observations = 400.

Table 3 summarizes the performance of the proposed method as compared with the state-of-art, when applied to the considered dataset. Better results are obtained as compared to the state-of-art owing to the novelty of the proposed algorithm that takes into consideration the degree of semantic relatedness of the candidate answer to the ideal answer decided and provided by the teacher/evaluator. This would in turn help in impartial evaluations of the answer sheets.

Hence, it can be concluded that the proposed method yields promising results. This can be attributed to the fact that the state-of-art doesn't take into consideration the semantic relationships and lexical expansion, but the proposed method does. It should also be highlighted here that IndusMarker [24] generates the word cloud in an automated manner which is to be manually analyzed by the evaluator. The proposed system on the other hand

Table 3 Standard deviation for accuracy and time for proposed method vs. state-of-art methods when tested on synthetic dataset

S. No.	Parameter	IndusMarker [24]	Superlative model [25]	Fuzzy WordNet graph-based keyword selection [26]	Proposed method
1	RMSE	0.367	0.345	0.33	0.319
2	Time (Seconds)	5.96	4.78	4.31	3.12

generates the WordNet graphs and assigns the scores automatically. This in turn assists in reducing the time of evaluation which is another significant aspect of answer sheet checking. In order to further increase the accuracy, there is a need to incorporate more measures of semantic relatedness.

5 Conclusion and Future Scope

This paper proposes a novel concept for answer sheet evaluation using the concept of text similarity applied to WordNet graphs. The answer sheets are evaluated by identifying the common nodes that occur between the node set of the ideal answer WordNet graph and the candidate answer WordNet graph. This kind of an evaluation combines the various significant concepts related to text similarity like semantic and structural dependencies. The root mean square error for the proposed approach was found to be 0.319 when tested on a dataset consisting of 400 students answer sheets. Unlike the state-of-art, the proposed method generates the WordNet graphs and assigns the scores automatically which in turn assists in reducing the time of evaluation. This shows that the proposed approach of answer sheet evaluation yields promising results in terms of both accuracy and time of evaluation. This work is suitable in scenarios where the student enters the correct spelling of the concerned words. The WordNet graph for erroneous non-words won't be generated. In the future, this work might be extended to incorporate measures to resolve this issue. Although marks are deducted in manual evaluation too for incorrect spellings, but some partial assignment is possible.

References

1. Li, X., Liu, N., Yao, C. L., & Fan, F. L. (2017). Text similarity measurement with semantic analysis. *International Journal of Innovative Computing Information and Control., 13*(5), 1693–1708.
2. Li, X., Yao, C. L., Fan, F. L., & Yu, X. Q. (2017). A text similarity measurement method based on singular value decomposition and semantic relevance. *Journal of Information Processing Systems, 13*(4), 863–875. https://doi.org/10.3745/jips.02.0067.
3. Al-Smadi, M., Jaradat, Z., Al-Ayyoub, M., & Jararweh, Y. (2017). Paraphrase identification and semantic text similarity analysis in Arabic news tweets using lexical, syntactic, and semantic features. *Information Processing and Management, 53*(3), 640–652. https://doi.org/10.1016/j.ipm.2017.01.002.
4. Kchaou, D., Bouassida, N., & Ben-Abdallah, H. (2017). UML models change impact analysis using a text similarity technique. *IET Software, 11*(1), 27–37. https://doi.org/10.1049/iet-sen.2015.0113.
5. Cho, S. G., & Kim, S. B. (2017). A data-driven text similarity measure based on classification algorithms. *International Journal of Industrial Engineering-Theory Applications and Practice, 24*(3), 328–339.

6. Abdul-Jabbar, S. S., & George, L. E. (2017). A comparative study for string metrics and the feasibility of joining them as combined text similarity measures. *ARO The Scientific Journal of Koya University, 5*(2), 6–18. https://doi.org/10.14500/aro.10180.
7. Reddy, G. S., & Rajinikanth, T. V. (2017). A text similarity measure for document classification. *IADIS-International Journal on Computer Science and Information Systems, 12*(1), 14–25.
8. Abdul-Rahman, A., Roe, G., Olsen, M., Gladstone, C., Whaling, R., Cronk, N., et al. (2017). Constructive visual analytics for text similarity detection. *Computer Graphics Forum, 36*(1), 237–248. https://doi.org/10.1111/cgf.12798.
9. Atoum, I., & Otoom, A. (2016). Efficient hybrid semantic text similarity using WordNet and a corpus. *International Journal of Advanced Computer Science and Applications, 7*(9), 124–130.
10. Bao, X. A., Dai, S. C., Zhang, N., & Yu, C. H. (2016). Large-scale text similarity computing with spark. *International Journal of Grid and Distributed Computing, 9*(4), 95–100. https://doi.org/10.14257/ijgdc.2016.9.4.09.
11. Kashyap, A., Han, L., Yus, R., Sleeman, J., Satyapanich, T., Gandhi, S., et al. (2016). Robust semantic text similarity using LSA, machine learning, and linguistic resources. *Language Resources and Evaluation, 50*(1), 125–161. https://doi.org/10.1007/s10579-015-9319-2.
12. Rahutomo, F., & Aritsugi, M. (2014). Econo-ESA in semantic text similarity. *Springerplus*. https://doi.org/10.1186/2193-1801-3-149.
13. Huang, C. H., Liu, Y., Xia, S. Z., & Yin, J. A. (2011). A text similarity measure based on suffix tree. *Information-an International Interdisciplinary Journal, 14*(2), 583–592.
14. Quan, X. J., Liu, G., Lu, Z., Ni, X. L., & Wenyin, L. (2010). Short text similarity based on probabilistic topics. *Knowledge and Information Systems, 25*(3), 473–491. https://doi.org/10.1007/s10115-009-0250-y.
15. Sun, Z. H., Errami, M., Long, T., Renard, C., Choradia, N., & Garner, H. (2010). Systematic characterizations of text similarity in full text biomedical publications. *PLoS ONE, 5*(9), 11. https://doi.org/10.1371/journal.pone.0012704.
16. Atlam, E. (2008). A new approach for text similarity using articles. *International Journal of Information Technology & Decision Making, 7*(1), 23–34. https://doi.org/10.1142/s021962200800279x.
17. Lewis, J., Ossowski, S., Hicks, J., Errami, M., & Garner, H. R. (2006). Text similarity: An alternative way to search medline. *Bioinformatics, 22*(18), 2298–2304. https://doi.org/10.1093/bioinformatics/btl388.
18. Liu, T., & Guo, J. (2005). Text similarity computing based on standard deviation. *Advances in Intelligent Computing, 3644,* 456–464.
19. Ozalp, S. A., Ulusoy, O. (2005). Effective early termination techniques for text similarity join operator. In Proceedings of computer and information sciences—ISCIS 2005 (Vol. 3733, pp 791). Berlin: Springer
20. Navigli, R., & Lapata, M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(4), 678–692.
21. Jain, A., Mittal, K., & Tayal, D. K. (2014). Automatically incorporating context meaning for query expansion using graph connectivity measures. *Progress in Artificial Intelligence, 2*(2–3), 129–139.
22. Jain, A., Tayal, D. K., & Vij, S. (2017). A semi-supervised graph-based algorithm for word sense disambiguation. *Global Journal of Enterprise Information System, 8*(2), 13–19.
23. https://wordnet.princeton.edu/
24. Siddiqi, R., Harrison, C. J., & Siddiqi, R. (2010). Improving teaching and learning through automated short-answer marking. *IEEE Transactions on Learning Technologies, 3*(3), 237–249.
25. Jayashankar, S., & Sridaran, R. (2017). Superlative model using word cloud for short answers evaluation in eLearning. *Education and Information Technologies, 22*(5), 2383–2402.
26. Vii, S., Tayal, D., & Jain, A. (2019). A fuzzy WordNet graph based approach to find key terms for students short answer evaluation. In 2019 4th international conference on internet of things: Smart innovation and usages (IoT-SIU) (pp 1–6). IEEE
27. Sijimol, P. J., & Varghese, S. M. (2018). Handwritten short answer evaluation system (HSAES).
28. Van Hoecke, O. D. C. S. (2019). Summarization evaluation meets short-answer grading. In Proceedings of the 8th workshop on NLP for computer assisted language learning, pp. 79–85
29. Roy, S., Rajkumar, A., & Narahari, Y. (2018). Selection of automatic short answer grading techniques using contextual bandits for different evaluation measures. *International Journal of Advances in Engineering Sciences and Applied Mathematics, 10*(1), 105–113.

Sonakshi Vij is a research scholar pursuing Ph.D. in Computer Science Engineering from Indira Gandhi Delhi Technical University for Women, Delhi in the field of Natural Language Processing. She has also worked as a guest faculty there. She qualified GATE and did her masters (M.Tech) in Information Security from Ambedkar Institute of Advanced Communication Technologies and Research, under Govt. of NCT Delhi affiliated to Guru Gobind Singh Indraprastha University, Delhi, India. She has completed her bachelor's degree (B.Tech) in Computer Science Engineering from BPIT, GGSIPU, Delhi. In the very early stage of her academics career she has published more than 20 research papers in International journals/conferences with proceedings in reputed publications like ACM, IEEE, Springer etc. Her research areas include Natural Language Processing, Fuzzy Logic, Soft Computing, Pattern Recognition, Information Retrieval, Sentiment Analysis and Machine Learning. She is a member of IAENG Society of Artificial Intelligence, Computer Science, Data Mining, Scientific Computing and Software Engineering.

Prof. Devendra Tayal is B.Sc (Hons.)(Maths), M.Sc (Maths), M.Tech (Computer Science & Technology) and Ph.D. (Computer Science & Technology). He is presently working as a Professor in the Department Computer Science & Engineering, Indira Gandhi Delhi Technical University for Women (earlier known as IGIT, GGSIP University, Kashmere Gate, Delhi), Govt. of Delhi, India and also served as Head of Department (Computer Science and Engineering) from 2007 to 2013, in the same university. He did his Ph.D. in the field of Fuzzy Databases on the topic "Data Dependencies using Fuzzy Functions in Fuzzy Relational Databases" and published many research papers in well-known International Journals with high impact factor including Fuzzy Sets & Systems, Journal of Uncertainty, Fuzziness and Knowledge based systems, Expert Systems with Applications, International Journal of Applied Intelligence etc. He has published more than 70 research papers with most of them in well-known international journals published by Elsevier, World scientific, Springer Verlag and IEEE conferences etc. His research interests include Fuzzy Logic based systems, Fuzzy Databases, Type-2 Fuzzy sets, Fuzzy based Information retrieval systems, Sentiment Analysis etc. He is a member of International Advisory committee of International Journal of Computer Science, Hongkong and member of International Advisory Board of International Journal of Software Engineering & Applications, Korea. Besides this, he is a referee on Editorial board of various International Journals including IEEE Transactions of Fuzzy Systems. He has extensively delivered lectures in Conferences, Seminars and Workshops and has also conducted many Conferences & Workshops.

Dr. Amita Jain is B.E., M. Tech. (Information Technology) and Ph.D. (Computer Science & Technology). She did her Ph.D. in Natural Language Processing and Fuzzy Logic from Jawaharlal Nehru University. She is presently working as an Assistant Professor in the Department Computer Science & Engineering at Ambedkar Institute of Advanced Communication Technologies and Research, Govt. of NCT of Delhi, India. She has published more than 75 research papers in highly reputed International Journals and conferences including ACM Transactions, IEEE, Springer etc. Her research interests include Fuzzy Logic based intelligent systems, Information Retrieval, Natural Language Processing, Sentiment Analysis etc. She has extensively delivered talks in International Conferences, Seminars and Workshops etc. She has also chaired special sessions in various reputed international conferences.