DOI: https://doi.org/10.24297/ijrem.v10i0.8495

Automated Answer Scoring for Engineering's Open-Ended Questions

Muhammad Sohail Ahmed, Ph.D.

118 Sill Hall, Eastern Michigan University. Ypsilanti, MI. USA

mahmed6@emich.edu

Abstract

Audience Response System (ARS), like "clicker," has proven their effectiveness in students' engagement and in enhancing their learning. Apart from close-ended questions, ARS can help instructors to pose open-ended questions. Such questions are not scored automatically for that Automated Text Scoring; ATS is vastly used. This paper presents the findings of the development of an intelligent Automated Text Scoring, iATS, which provides instantaneous scoring of students' responses to STEM-related factual questions. iATS is integrated with an Audience Response System (ARS), known as iRes, which captures students' responses in traditional classrooms environment using smartphones. iATS Research is conducted to code and test three Natural Language Processing (NLP), text similarity methods. The codes were developed in PHP and Python environments. Experiments were performed to test Cosine similarity, Jaccard Index and Corpus-based and knowledge-based measures, (CKM), scores against instructor's manual grades. The research suggested that the cosine similarity and Jaccard index are underestimating with an error of 22% and 26%, respectively. CKM has a low error (18%), but it is overestimating the score. It is concluded that codes need to be modified with a corpus developed within the knowledge domain and a new regression model should be created to improve the accuracy of automatic scoring.

Keywords: Audience Response System, Close-Ended Questions, Automated Text Scoring, Automated Essay Evaluation, Natural Processing Language

Introduction

Creating active learning and improve participation and engagement in a sizeable live classroom is challenging. In small class sizes, too, encouraging all students to participate and engage in class become difficult with the presence of shy or non-participating students, [1,2]. Audience Response System (ARS), like clickers, has provided a tool that has proved to be very useful in such an environment, [3]. It is very well documented how clicker has helped in creating an active learning environment and improving students' participation and engagement, [4, 5, 6]. Such a meeting includes students responding to a question in live class and been evaluated by the system or instructor immediately. ARS works very well with the closed-ended questions by providing the grades immediately. In the open-ended question, however, instructors need to manually grade and provide comments on students' responses, which depletes a significant class time, [7]. With the advancement of technology and inclusion of mobile learning (m-learning) in the academic environment, clicker and similar ARS has become a costly and inefficient tool, [8, 9]. In today's' digital world, most of the commercial ARS also allows students to participate using laptops, mobile devices, or iClicker remotes. ARS uses websites and Apps to communicate with instructors. These online ARS grade students' responses to close-end questions only, and illustrate results as bar charts to create discussion in the class. Ahmed, [10], describe the model of an online ARS that give students the capability to use their smartphones to respond to instructor question, using SMS and web-messaging. This eliminates the cost of buying any ARS hardware.

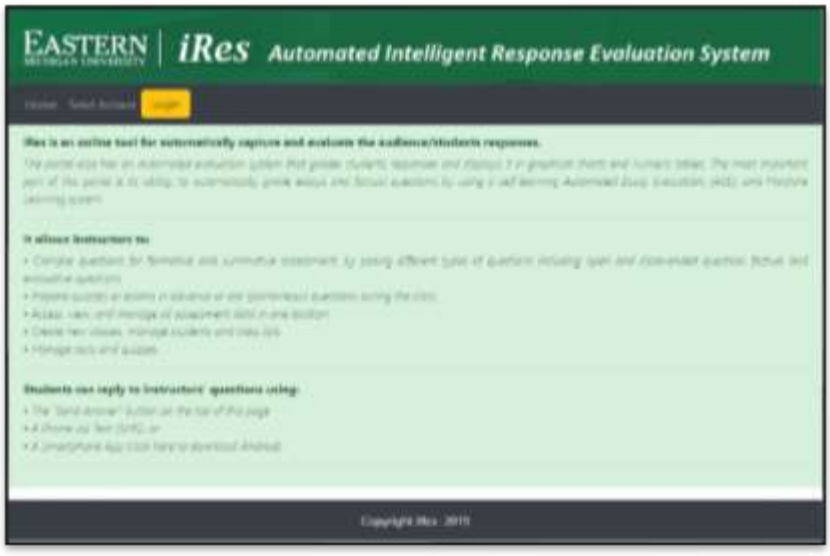
Automated Essay Scoring (AES), also called Automated Essay Evaluation (AEE) or Automated Text Scoring, (ATS), has been around since the '60s, [10]. Automated Text Scoring (ATS) provides a cost-effective and consistent alternative to human marking. Most of the AES in the market aims to evaluate essays on many social, historical, and political topics. Their use is widespread in high stake assessments like GRE, GMAT, TOEFL, SAT, etc. [7]. E-

rater was the first system to be deployed in high-stake evaluation in 1999, [10], and since then, it got a great deal of attention from the rhetoric and composition/writing studies community, [11]. Several researchers have been arguing about the value add and effectiveness of AEE. Kane, [12], explains that both human grading and AEE scoring are pruned to controversy as they relate to a "number/ grade" or a "statistical artifact," respectively, instead of accessing the essay's complex information. Condon, [11], argues that AEE only provides minor relief for teaching writing. He claims that what the code does is just drudgery by dealing with grammar and mechanics. According to Peter Greene, [13], of Forbes, the biggest problem with such robo-grading, continues to be the algorithm's inability to distinguish between quality and drivel of essays.

The unavailability of an ARS that can also automatically grade responses for the open-ended questions is the main objective of this paper. It requires integrating ATS and ARS. Most of the available ATS/AEE targets the evaluation of essays. To evaluate and score short responses may require a different approach. This research focused on scoring STEM-related open-ended factual questions using Natural Language Processing, NLP.

Natural Language Processing (NLP), helps in developing applications to understand human languages, like automatically grading texts and sentences ATS software utilizes NLP methodology to simulate a qualitative aspect of human rater's scores, [14]. With the improvement in the computation power, since the '80s Machine Learning algorithm has dominated the processing in NLP, where it analyses corpus linguistics or real-world text samples for language processing using statistical inference to learn new rules. The fundamental concepts of NLP differ from those of Machine Learning or Software Engineering in general. Not all AEE can evaluate every aspect of language for grading purposes [15]. The most common elements of NLP focuses on grammar, usage, mechanics, style, organization, development (defined as the length of sentences), positive features (use of prepositions and the basic concept mapping of essential vocabulary), lexical complexity, (use of longer, polysyllabic words), in topic-specific prompts, and, topic-specific vocabulary usage, [16].

This research utilizes an online ARS called iRes (I Response) developed using the Ahmed, [10], ARS architecture. iRes is a web-based ARS that helps instructors to pose questions to the students in live classes, Figure#1. These questions can be essay based. Students can respond in multiple ways; by simple "text" message (SMS), or by web-messaging using a smartphone app or by using a website.



Figure#1. iRes an Audience Response System Website

The paper describes the development and testing of an ATS, called iATS. The iATS is integrated with iRes. iATS utilize Machine Learning and Data Science concepts to develop and test three NLP models to check similarities

between two texts. These NLP similarity methods are coded using Python and PHP. These codes use Natural Language Tool Kit (NLKT), a python library, and NPLTools deals with PHP.

Material and Method

Research Problem

In Science, Technology, Engineering, and Mathematics (STEM) education posing questions regarding laws, principles, or known facts is a common way to ensure students' ability to comprehend. Students' responses to such questioning are a short essay. These short essays are essential testing tools for assessing students' academic achievement, their ability to integrate ideas, and the ability to apply the facts in solving real-world problems. As with any other type of essay, manually evaluating and scoring such short essays is time-consuming. In a live classroom, such questions can be asked using ARS. As identified above in the introduction, all existing automatic scoring of essays or texts is currently useful for non-STEM related academic fields. At the same time, the current Audience Response Systems do not have the ability to pose essay –type questions other than to create classroom discussions. Similarly, there is no Automatics Scoring system available that work with any ARS to score these responses automatically.

This research is a part of the development of a project on the Audience Response System that would use Machine Learning and NLP to score students' responses to the STEM-related factual question automatically. The project is called Automated Intelligent Response Evaluation System, AiRes, which is integrated into an Audience Response System, called iRes. iATS has two development phases. This paper describes phase one, which includes the development of the automated scoring system using NLP, features, and calculating similarity scores using three different NLP similarity methods.

Methodology

Data is collected using iRes, where the instructor first compiles a question and provide a "golden answer" along with some keywords. The golden answer given by the instructor is treated as one of the best solutions. The students' responses are captured through SMS or web-messages and are graded by the instructor. These are then stored in the iRes database, Figure #2.

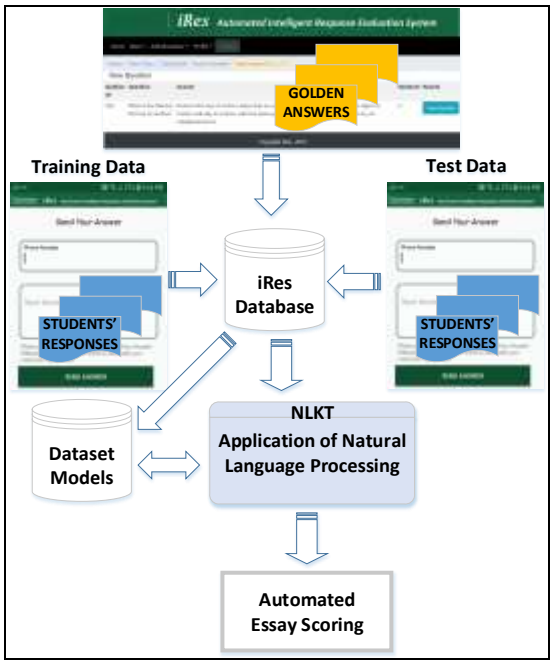


Figure #2. iRes and iARS Integration Model

Figure #3 illustrate the research's basic block diagram level implementation methodology. First, NLP features are extracted for the Instructor's "gold answer" and students' responses, to conduct statistical and other analysis. Finally, responses are evaluated by computing NLP similarity between "gold answer" and students' responses using Cosine Similarity, Jaccard Similarity, and Corpus-based and knowledge-based measures, (CKM) methods.

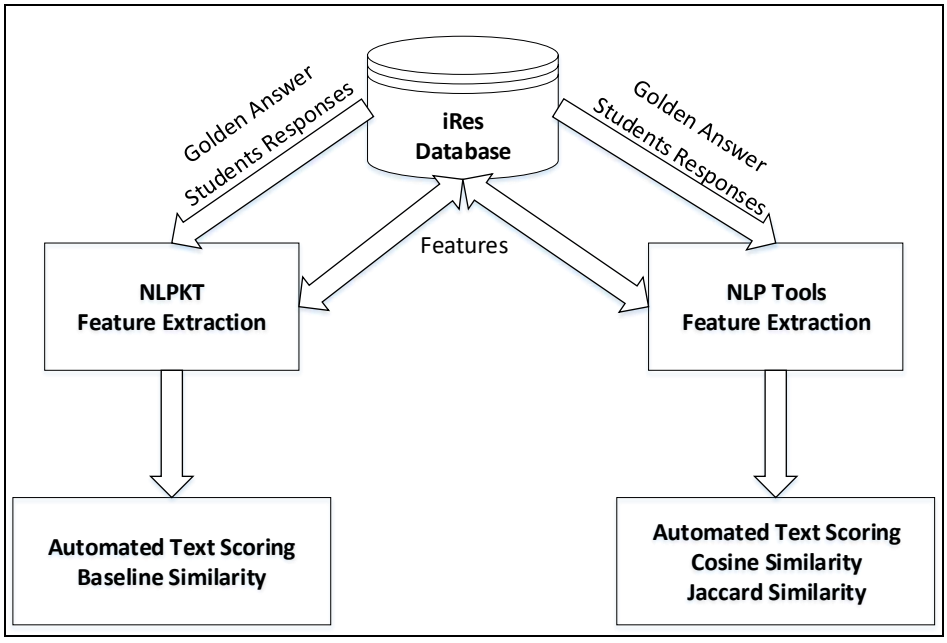


Figure #3: iATS Implementation Methodology

Code and Dataset

Python 3.x and PHP 7.2 were used to develop the iATS since there are multiple libraries already available for working with natural language processing. Natural Language Toolkit (NLTK), which is the most critical NLP library written in Python and Nlp Tools, is a library for natural language processing written in PHP. One similarity code for iATS was developed using Python, and two were generated using Nlp Tools [17, 18]. NLP applications need a dataset to compute the similarity. These codes employed Semantic Textual Similarity or STS Benchmark and the SICK, (Sentences Involving Compositional Knowledge). These two are widely used datasets to compute similarity. STS Benchmark comprises a selection of the English datasets used in the STS tasks organized in the context of SemEval between 2012 and 2017. While SICK is a data set for compositional distributional semantics. SemEval (Semantic Evaluation) is an ongoing series of evaluations of the computational semantic analysis system.

Features

Attali & Burstein, [19], reported twelve individual elements that reflect essential characteristics in essay writing that aligned with human scoring criteria. iATS extracted the Statistical/ Numerical Features (like features like the total word count and sentence count, average sentence length, paragraph count per response), Orthography, Bag of Words (BoW), and Parts of Speech.

Similarity:

There are various methods to find the semantic similarity in meaning between two sentences. In this research, we have used the following three ways:

1) Cosine Similarity converts texts into vectors and calculates similarity by computing the cosine of the angle between the two vectors, [20]. The cosine similarity, cos(θ), is represented using a dot product and magnitude as

$$similaty = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A^2}\sqrt{\sum_{i=1}^{n} B^2}} \qquad \text{(Equation 1)}$$

Where Ai and Bj are the components of Gold-answer and response vectors, respectively.

2) Jaccard Similarity uses the root words of the two texts and computes the similarity between them through their intersection divided by union. Jaccard is actually counting the similar and non-similar words in the gold-answer, X, and student's response, Y.

$J(X,Y) = |X \cap Y| / |X \cup Y|$ (Equation 2)

The Jaccard Index is dependent on a unique set of words for every sentence; therefore, duplication will not affect it, [21].

3) The third method utilizes an aspect of Corpus-based and knowledge-based measures of similarity (CKM), developed by Mihalcea, Corley & Strapparava, [22]. It uses external resource WordNet®. WordNet® is an extensive lexical database of English where every part of speech (pos) noun, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. This similarity code is developed in Python using NLKT. The only difference between this code and corpus-based and knowledge-based measures of similarity method is the fact that in this code, the max similarity is not weighted with an Inverse-Document-Frequency (IDF), [19].

Results and Discussion

Testing was performed in a freshman introductory mechanics class using a simple physics question; "What is Newton's first law of motion?" The question was posed using iRes, and student's responses were captured and were automatically graded by the iATS. Thirty-five students participated in the test. Figure#4 illustrates the outcome of the iRes result page.

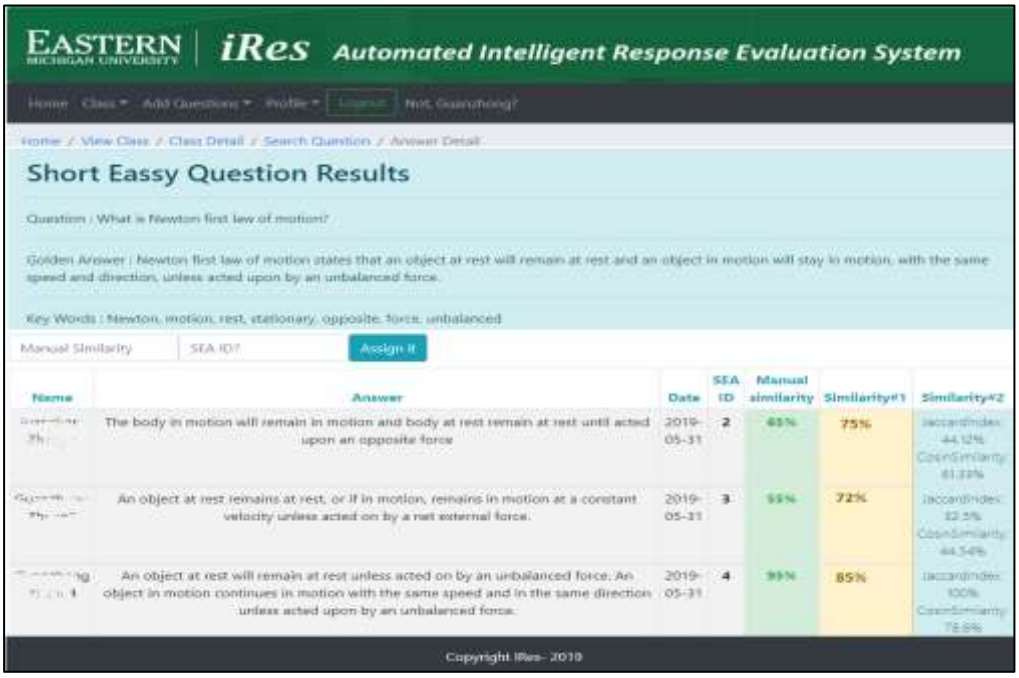


Figure #4: The iRes page showing the similarity results

The responses were then graded by the Instructor. Table #1 document these results.

Student Name	Similarity CKM	Cosine Similarity	Jaccard Index	Manual Scores	Error (Jaccard-Man)	Error (Cosine-Man)	Error (Similarity - Man)
1	75%	61%	44%	70%	-37%	-13%	7%
2	72%	44%	32%	65%	-51%	-32%	11%
3	85%	78%	100%	95%	5%	-18%	-11%
4	88%	44%	60%	75%	-20%	-41%	17%
5	96%	53%	74%	80%	-8%	-34%	20%
6	92%	55%	71%	85%	-16%	-35%	8%
7	89%	60%	50%	75%	-33%	-20%	19%
8	93%	67%	71%	85%	-16%	-21%	9%
9	90%	50%	74%	80%	-8%	-38%	13%
10	92%	48%	81%	75%	8%	-36%	23%
35	..	..	..	..	..	..	..
			Average Errors		**-22%**	**-26%**	**18%**

Table #1. Data for Similarity Analysis

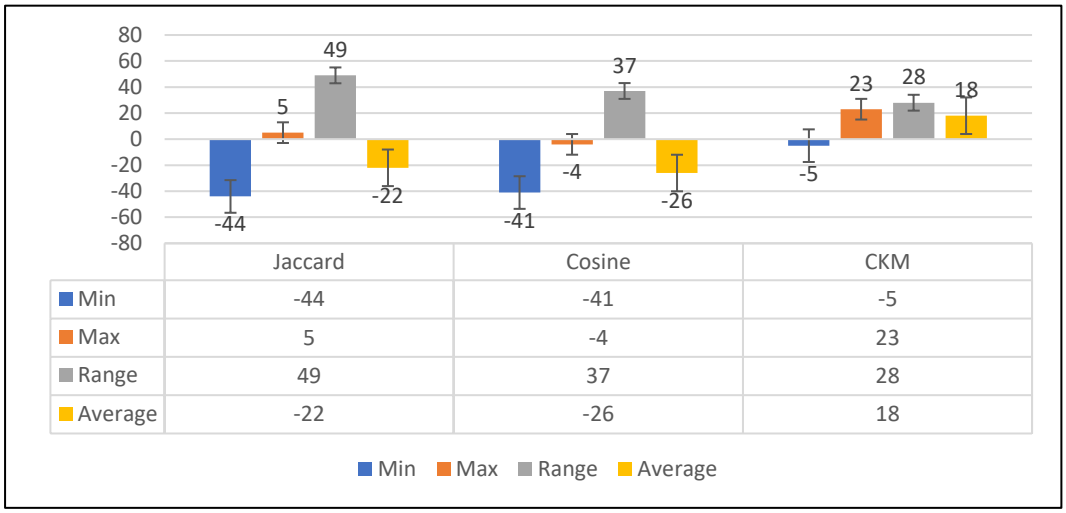
The results suggest that

a) Error for Jarrard Index ranges between -44% and 5%.

b) Error for Cosine ranges between -41% and -4%.

c) Error from Similarity (CKM) ranges between -5% and 23%.

d) CKM method predicted the scores with the least error.



Figure#5. Data Analysis for Similarity Scores

e) Cosine is always underestimating, while Jarrard Index is mostly underestimating compared with CKM, which is mostly overestimating the scores, Figure #5.

f) Individual data scores illustrate Cosine and Jaccard Index similarity scores are not good predictor as the error with manual scores are all greater than 20 %.

Conclusions

The research tested the applicability of an automated response scoring system with an ARS. The results obtained from testing three similarity methods are exciting but not unusual.

Cosine similarity error can be explained by the fact that this method is sensitive to the total length of vectors compose of features and is effected by numbers of similar words in the sentences. Until and unless the two sentences, student's response and golden-answer, are identical and have multiple similar words, the Cosine similarity method will not give a higher similarity number. The number of similar words increases the cosine similarity increases. It is therefore hypothesized that if keywords are utilized in Cosine similarity, it will increase its accuracy.

The Jaccard Index similarity error can be linked to the fact that the Jaccard Index is only considering a unique set of words in student response and golden-answer. Therefore, duplication of the word will not affect the Jaccard Index. This can explain why the Jaccard Index has a more significant error than Cosine similarity as it is not counting the multiple similar words in a sentence numerous times. The other reason for a higher error in the Jaccard method can be the fact that the sentence lengths are small in most of the responses.

It was expected that CKM method scores would be closer to manual scores, but it did not end up that way. It is assumed that this can be the result of one of the following or a combination of them:

i) The max similarity is not weighted with an Inverse-Document-Frequency (IDEF) and, it was not included in the code.

ii) The fact that in WordNet®, some have issues with calculating the similarity between adjectives and adverbs.

iii) The corpus used is not matching with the content been tested. Being a STEM-related text it is possible that a separate corpus is needed

The next step in this project would be to work on improving the accuracy of the automated scores. The CKM and Cosine need to be modified, or a different method is required in order to obtain better results.

Data Availability (excluding Review articles)

The author can be contacted to get a copy of the data presented.

Funding Statement

The research was funded through EMU's 2019 Summer Research Activity Award.

References

1. Mayer, R. E., Stull, A., Deleeuw, K., Almeroth, K., Bimber, B., Chun, D., … Zhang, H. (2009). Clickers in college classrooms: Fostering learning with questioning methods in large lecture classes. Contemporary Educational Psychology, 34(1), 51–57. doi: 10.1016/j.cedpsych.2008.04.002

2. Wittrock, M. C. (1990). Generative processes of comprehension. Educational Psychologist, 24, 354–376.

3. Mayer, R. E. (2008). Learning and instruction. New York: Pearson Merrill Prentice Hall.

4. Duncan, D., 2005. Clickers in the Classroom: How to Enhance Science Teaching Using Classroom Response Systems. Addison-Wesley, New York.

5. Herreid, C., 2006. ''Clicker" cases: introducing case study teaching into large classrooms. Journal of College Science Teaching 36 (2), 43–47.

6. Kane, M. T. (2006). Validation. In: R. L. Brennan (Ed.), Educational measurement (4th ed., pp. 17–64). Westport, CT: American Council on Education/Praeger.

7. Patterson, B., Kilpatrick, J., & Woebkenberg, E. (2010). Evidence for teaching practice: The impact of clickers in a large classroom environment. Nurse Education Today, 30(7), 603–607. doi: 10.1016/j.nedt.2009.12.008

8. Blood, Ian. "Automated Essay Scoring: A Literature Review". Working Papers in TESOL and Applied Linguistics 11.(2011) (2016): n. pag. Web. 16 Apr. 2016.

9. Cavus, N., Ibrahim, D. (2009) 'M-Learning: An experiment in using SMS to support learning new English language words', British Journal of Educational Technology, 40(1): 78-91.

10. Ahmed, M. S. (2017). Lessons Learned from NSF I-Corps Boot Camp, Journal of Education and Practice, 8(26) pg 1-10. http://www.iiste.org/Journals/index.php/JEP/article/view/38894.

11. Zupanc, K., & Bosni´c, Z. (2015). Advances in the Field of Automated Essay Evaluation. Informatica, 39, 383–395.

12. Condon, W. (2013). Large-scale assessment, locally-developed measures, and automated scoring of essays: Fishing for red herrings? Assessing Writing, 18(1), 100-108. doi:10.1016/j.asw.2012.11.001

13. Greene, P. (2018, July 4). Automated Essay Scoring Remains An Empty Dream. Retrieved from https://www.forbes.com/sites/petergreene/2018/07/02/automated-essay-scoring-remains-an-empty-dream/#fba935174b91

14. Leacock, C., Chodorow, M., Gamon, M., & Tetreault, J. (2010). Automated grammatical error detection for language learners. In E. Hirst, Synthesis Lectures on Human Language Technologies, San Rafael, CA: Morgan & Claypool Publishers.

15. Shermis, M. D., & Burstein, J. C. (2013). Handbook of automated essay evaluation. New York, NY: Routledge.

16. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. Beijing: OReilly.

17. Sphinx, (nd) Natural Language Toolkit. Retrieved August 21, 2019, from https://www.nltk.org/.

18. Trilla, A. (2012, February 28). Natural Language Processing Toolkit for PHP. Retrieved from http://nlptools.atrilla.net/doc/html/

19. Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater V.2. Journal of Technology, Learning, and Assessment, 4 (3), 1-21.

20. Contreras, J. O., Hilles, S., & Abubakar, Z. B., (2018). Automated Essay Scoring with Ontology based on Text Mining and NLTK tools. 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE). doi: 10.1109/icscee.2018.8538399

21. Das, K., & Sinha, S. K. (2019). Identification and Analysis of Future User Interactions Using Some Link Prediction Methods in Social Networks. Data, Engineering, and Applications, 83–94. doi: 10.1007/978-981-13-6347-4_8

22. Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and Knowledge-based Measures of Text Semantic Similarity. *Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, *25*. Retrieved from https://www.aaai.org/Papers/AAAI/2006/AAAI06-123.pdf