

# Analysis of Knowledge Testing Results Using the Vectorization of Sentences

Chulpan Minnegalieva<sup>a</sup>, Gulshat Sabitova<sup>b</sup> and Almaz Gayaliev<sup>c</sup>

*Institute of Computational Mathematics and Information Technologies, Kazan Federal University, Kremlyovskaya street  
18, Kazan, 420008, Russian Federation  
mchulpan@gmail.com, gulshat.9922@gmail.com, gayaliev.almaz@mail.ru*

**Keywords:** Machine Learning, Vector Model, Word Embedding, Word2vec, Doc2vec, Cosine Similarity, Binary Classification.

**Abstract:** The paper discusses the pre-assessment of students' answers to test questions using word embedding. We trained word2vec and doc2vec models on computer graphics texts. Students were enrolled in a course called "Computer Graphics and Design". They gave answers to questions before the course, during the course, immediately after the completion of the course, and one month later. Students gave answers consisting of one or more sentences. First, we identified the correct and incorrect answers. Then we compared the students' answers to the right answers using cosine similarity. The results showed that clearly incorrect answers can be identified using the word2vec and doc2vec models. For answers that are close to right answers, we suggest doing an additional word-check for words that are meaningful to the question. The same check can be performed if the cosine measure has taken boundary values. A binary classification of answers was carried out, indicators were studied. The results showed that using the classification, it is possible to identify with sufficient accuracy the incorrect answers.


## 1 INTRODUCTION


Knowledge testing usually includes questions that students can answer in a free form. Checking the answers to the questions takes a long time for teachers. Using modern approaches to text analysis, it is possible to propose a method for preliminary assessment of students' answers.


The tasks of news categorization, text sentiment analysis, methods for improving the quality of translation are currently being solved, including using machine learning methods. In recent years, many deep-learning-based models for text classification have been developed (Minaee, 2022). For example, the methods of sentiment categorization systems were classified by the authors in the paper and compared their advantages and disadvantages (Wankhade, 2022).

Tomas Mikolov proposed an approach to word embedding, which he called word2vec (Mikolov, 2013). After presenting the words in vector form, cosine similarity calculation method is used to determine the similarity value. The authors shown in their analysis how the various hyperparameters influence learning differently (Di Gennaro, 2021). This approach is used today in various text analysis problems. The authors suggest to use word embeddings to look for substitute words used for academic writing. A word embedding model trained on a collection of academic articles in a specific domain (for example publications about computational linguistics and natural language processing) might suggest similar expressions that comply to that writing style and are suited to that domain (Goh, 2020). Usually each language is considered separately. It is also proposed to use the possibilities of texts written in other languages. The

---

<sup>a</sup>  <https://orcid.org/0000-0003-4648-1623>

<sup>b</sup>  <https://orcid.org/0009-0008-1840-4819>

<sup>c</sup>  <https://orcid.org/0009-0006-9735-675X>

paper describes a model that represents the sentiment and semantics of words and improves the effect of Vietnamese sentiment analysis model by using the English corpus (Huang, 2020).

In this paper, we analyze students' answers to test questions using word embedding.

## 2 ANALYSIS OF KNOWLEDGE TESTING RESULTS

### 2.1 Vectorization of Words and Sentences

We used vector word representations to analyze students' free-form answers. The students enrolled in a course called "Computer Graphics and Design". They gave answers to questions at different times, before and after the completion of the course. We trained the word2vec and doc2vec models using the Gensim library. To train the models, we used texts from textbooks on computer graphics. Data preprocessing is the important phase of a machine learning project (Kadam, 2022). During preprocessing, we got rid of punctuation and numbers, cleared the data from stop words, and conducted lemmatization. The morphological analyzer pymorphy2 was used to prepare the data.

In distributive semantics, words are represented as multidimensional vectors. Semantic similarity is calculated as cosine proximity between vectors of two words (Jatnika, 2019). To determine the vector of the sentence, we used the average value of the vectors of the words included in the sentence. When training the word2vec model, we checked different values of the window parameter – 1, 3, 5, 10. We found that when training models on our corpus of texts with a window value of 3, correct and incorrect answers are better distinguished. With a large window value, for example, 10, the vectors of words from computer graphics texts are close by cosine measure. And this makes it difficult to distinguish correct answers from incorrect and incomplete ones. We also tested the model with different values of the size and min\_count parameters. Changing the size parameter did not significantly affect the result. As the value of min\_count increases, fewer words remain in the model, the vectors of words used in the answers again turn out to be close in cosine measure. Therefore, when training the word2vec model, we took the value of the min\_count parameter equal to 1. The doc2vec model was also trained on information technology texts, including texts on computer graphics topics.

When training the doc2vec model, we also determined the necessary parameters.

### 2.2 Analysis of Answers

Note that for this article we have translated the questions asked to students and their answers into English.

We have collected the answers to 20 test questions. Then, using the word2vec model, the answer vectors were compared with the right answer vector. We tried different approaches. When analyzing the answers to some of the questions, the average vector of all correct answers given by students was taken as the right answer. When analyzing answers to other questions, we took the answer from the textbook as the right answer. Consider some answers to the question "Give a definition of the normal vector". (The normal is a vector perpendicular to the tangent plane of the surface at a given point). In Table 1, the cosine similarity shows how close the answer vectors are to the right answer vector.

Table 1: Cosine similarity of answer vectors.

	Answer	Cosine similarity
1	Any nonzero vector belonging to a line perpendicular to the tangent plane (correct answer).	0.889854
2	Perpendicular to the plane (incomplete answer).	0.465488
3	This is a vector perpendicular to the center of the plane (inaccurate answer).	0.485951
4	The position of the object to the plane is necessary for determination (incorrect answer).	0.275967

As you can see from Table 1, answer 4 is different from the right answer, the cosine value is also small. Answer 1 is closest to the right one both in content and in cosine value. Answers 2 and 3 are similar to the right answer, but we cannot say that they are completely correct.

#### 2.2.1 The Results of Training Models on Different Texts

40 students had to continue the sentence "Multimedia is... ". We found the cosine similarity between the vectors of students' answers and the vector of the right answer ("the combination of different content forms such as text, audio, images, animations, or video"). Two word2vec models were used for this. The first model was trained on a large corpus of common texts;

the second model was trained on computer graphics texts. Some answers and cosine values are shown in Table 2.

Table 2: Closeness of answers to the right answer.

Answer	Cosine similarity (common text)	Cosine similarity (special text)
Modern technologies that allow presenting information in audio, video, and text formats (correct answer).	0.844519	0.842886
Library of graphic files (incorrect answer)	0.502775	0.753846

After analyzing all the answers, we concluded that the two models rated the answers to some questions equally. But there were answers that the models rated differently. We assume that this depends on the wording of the question and on whether or not special terms were used.

### 2.2.2 Using Different Models

Let's consider another assignment. The students continued the sentence: "A graphic editor is...". We analyzed their answers with word2vec and doc2vec models. Both models were trained on computer graphics texts. Using the word2vec model, we found the sentence vector as the average of the word vectors. Then we found a cosine similarity between the vectors of students' answers and the vector of the right answer ("software for processing, creating, viewing, editing images using a computer"). The cosine values are shown in Table 3.

Table 3: Results of the analysis of answers using different models.

Answer	word2vec	doc2vec
Graphic information processing tool (correct answer).	0.735427	0.803608
Image processing software (correct answer).	0.964357	0,714491
I don't know (incorrect answer)	0.327918	0,606841

As you can see, the cosine values are different for the first two answers, but both models correctly classify the answers as correct. The vector of student's answer "I don't know" and the vector of correct answer are close (model doc2vec). Next, we analyzed the answers to another question "What can

be done with the Gradient tool in the graphics editor?" (one of the right answers: "gradual step-by-step change of color or intensity"). The cosine value between the vectors of the right answer and the answer "I don't know" in this case is -0.075676. We think that the answer to the question about the graphical editor contains more common words, the answer to the question about the Gradient tool contains more specific words. Therefore, in the second case, the answer "I don't know" was correctly identified by the model as incorrect. We plan to investigate this case further with other data.

### 2.2.3 Additional Checking of Answers

Students gave answers to the question "What does the gradient vector mean?" This is a variant of the correct answer: "a vector, with its direction indicating the direction of the fastest increase of a certain value". We calculated the values of the cosine between the vectors of student answers and the vector of the right answer. If the cosine value is greater than 0.5, we considered the answer was correct. Out of 31 answers, in 26 cases the word2vec model correctly determined whether the answer is correct or not. An example of a student's answer: "Maybe a direction". This answer is incorrect, the cosine value is 0.5638. Let's consider why such a result could be obtained. We have identified important words in the right answer. These are the words "direction", "increase". The student's answer also contains the word "direction". Therefore, the vectors turned out to be close, because we took the average of the vectors of words as the answer vector. When calculating the sentence vector, we tried to take the vectors of words that do not match the important ones with a different coefficient (for example, 10). In this case, the cosine value is 0.36878. If the answers are checked automatically, this answer will be correctly classified as incorrect. We suggest using this method for additional verification of the answers.

## 2.3 Binary Classification

Among the questions that students answered at different times were those that were easy or, on the contrary, turned out to be quite difficult for them. We have selected questions with a sufficient number of correct and incorrect answers. In the example discussed earlier, we determined the value of the threshold (0.5) ourselves. It was decided to evaluate the answers to the other question independently from the value of the threshold. We preliminarily distinguished correct and incorrect answers of

students, then applied a binary classification. For this we used the library scikit-learn (Python), a random forest model. In this case, when finding the sentence vector, the vectors of words included in them were taken with TF-IDF (term frequency – inverse document frequency) coefficients. For example, students had to continue the sentence "CDR files are used for... ". After analyzing the answers, we got Precision: 0.89, Recall: 0.86, F1-measure: 0.85, Accuracy: 0.86. We can assess the trend and continue checking with other data.

### 3 CONCLUSIONS

In this work, using the word2vec and doc2vec models, we compared students' answers to questions with the right answers. Texts on topics of computer graphics were used to train the models. Models identify clearly wrong answers with sufficient accuracy. When the answers are similar to the right ones, we suggest additional analysis. In this case, the words that are important for the question are compared. For questions that contained a balanced number of correct and incorrect answers, we used binary classification method. The classification determined the correct answers with sufficient accuracy. We plan to conduct a multiclass classification (identify correct, incorrect, inaccurate, incomplete, absurd answers). We also plan to develop an application based on the considered models for a preliminary analysis of the answers.

### ACKNOWLEDGEMENTS

This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program ("PRIORITY-2030").

### REFERENCES

- Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J., 2022. Deep Learning-based Text Classification: A Comprehensive Review. *ACM Computing Surveys* 54(3). 62:1–62:40.
- Wankhade, M., Rao, A.C.S., Kulkarni, C., 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review* 55(7), 5731–5780.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural*

- information processing systems* 26 (NIPS 2013), 3111–3119.
- Di Gennaro, G., Buonanno, A., Palmieri, F.A.N., 2021. Considerations about learning Word2Vec. *The Journal of Supercomputing* ,77(11), 12320–12335.
- Goh, C.L., Lepage, Y., 2020. An Assessment of Substitute Words in the Context of Academic Writing Proposed by Pre-trained and Specific Word Embedding Models. In: Nguyen, L.M., Phan, X.H., Hasida, K., Tojo, S. (eds) *Computational Linguistics. PACLING 2019. Communications in Computer and Information Science*, vol.1215. pp. 414–427. Springer, Singapore.
- Huang, Y., Liu, S., Qu, L., Li, Y., 2020. Effective Vietnamese Sentiment Analysis Model Using Sentiment Word Embedding and Transfer Learning. In: Qin, P., Wang, H., Sun, G., Lu, Z. (eds) *Data Science. ICPCSEE 2020. Communications in Computer and Information Science*, vol. 1258. pp. 36–46. Springer, Singapore.
- Kadam, V.P., Khandale, K.B., C., N.M., 2022. Text Analysis and Classification for Preprocessing Phase of Automatic Text Summarization Systems. In: Patel, K.K., Doctor, G., Patel, A., Lingras, P. (eds) *Soft Computing and its Engineering Applications. icSoft-Comp 2021. Communications in Computer and Information Science*, vol. 1572. pp. 382–396. Springer, Cham.
- Jatnika, D., Moch A.B., Arie A.S., 2019. Word2Vec Model Analysis for Semantic Similarities in English Words. *Procedia Computer Science* 157, 160–167.