Received: 26 January 2020 | Accepted: 19 April 2020

DOI: 10.1002/cae.22252

SPECIAL ISSUE ARTICLE

Weighted word embeddings and clustering-based identification of question topics in MOOC discussion forum posts

Aytuğ Onan[1] | Mansur Alp Toçoğlu[2]

[1]Department of Computer Engineering, Faculty of Engineering and Architecture, İzmir Katip Çelebi University, İzmir, Turkey

[2]Department of Software Engineering, Faculty of Technology, Manisa Celal Bayar University, Manisa, Turkey

Correspondence
Aytuğ Onan, Department of Computer Engineering, Faculty of Engineering and Architecture, İzmir Katip Çelebi University, 35620 İzmir, Turkey.
Email: aytug.onan@ikc.edu.tr

Abstract

Massive open online courses (MOOCs) are recent and widely studied distance learning approaches aimed at providing learning material to learners from geographically dispersed locations without age, gender, or race-related constraints. MOOCs generally enriched by discussion forums to provide interactions among students, professors, and teaching assistants. MOOC discussion forum posts provide feedback regarding the students' learning processes, social interactions, and concerns. The purpose of our research is to present a document-clustering model on MOOC discussion forum posts based on weighted word embeddings and clustering to identify question topics on discussion posts. In this study, four word-embedding schemes (namely, word2vec, fastText, global vectors, and Doc2vec), four weighting functions (i.e., term frequency-inverse document frequency [IDF], IDF, smoothed IDF, and subsampling function), and four clustering algorithms (i.e., K-means, K-means++, self-organizing maps, and divisive analysis clustering algorithm) for document clustering and topic modeling on MOOC discussion forum posts have been evaluated. Twenty different feature representations obtained from word-embedding schemes and weighting functions have been obtained. The feature representation schemes have been evaluated in conjunction with four clustering methods. For the evaluation task, the empirical results for the latent Dirichlet allocation have been also included. The empirical results in terms of adjusted rand index, normalized mutual information, and adjusted mutual information indicate that weighted word-embedding schemes combined with clustering algorithms outperform the conventional schemes.

KEYWORDS

discussion forums, document clustering, massive online open courses, text mining

1 | INTRODUCTION

Over the past decade, massive open online courses (MOOCs) have become increasingly popular and have enabled new ways to provide learning materials to learners from geographically dispersed locations without age, gender, or race-related limitations [22]. MOOCs can be defined as online courses offered to unlimited number of participants. The number of people participating in MOOCs has steadily increased, and there are several websites, such as Coursera, edX, MiríadaX or FutureLearn, which give thousands of participants free open courses from prestigious institutions around the world [31]. Usually, MOOCs are distinguished by free registration and provide convenient access to educational information and resources. Using MOOCs provides instructors with the opportunity to reach many students worldwide and provides students with the opportunity to reach a wide range of courses offered by instructors from prestigious higher institutions, which otherwise may not be possible [23]. MOOCs support the conceptualization of continuous professional learning and personalized lifelong learning [14]. Furthermore, MOOCs are means of providing a more accessible and democratized higher education model [25].

MOOC courses typically utilize traditional learning materials, such as short video tutorials, slide presentations, reading texts, problem sets, live chat, and online learning assessments [20]. In contrast to traditional classroom environments, almost all relevant communication in massive online open courses takes place on the MOOC discussion forums. Discussion forums are the only channel for most students to seek answers to queries, either provided by instructors, teaching assistants, or other participants. In traditional classroom environments, informal discussions with classmates constitute an important part of learning process. This, however, has been replaced by discussion forums on MOOCs. Hence, discussion forum posts provide potentially useful information regarding the learning processes on MOOCs. MOOC discussion forum posts provide feedback regarding the students' learning processes, social interactions, and concerns [51].

Educational data mining (EDM) is a new field of research concerned with the application of tools and techniques from data mining, machine learning, and statistics to data obtained from the educational domain, to better understand students and the settings of the learning process [42]. In EDM, techniques from machine learning and statistics can be employed on data from the educational domain to provide educational policymakers with useful insights to improve teaching and learning efficiency and quality [36]. EDM and learning analytics can be employed to address several learning problems on self-learning behavior, self-assessment, self-regulated learning, learning material evaluation, students' learning evaluation, assessment and monitoring, dropout and retention modeling, and data-driven decision making [3]. The main techniques of EDM include classification, clustering, visual data mining, statistics, association rule mining, regression, sequential pattern mining, and text mining [40].

Clustering (also known as cluster analysis) is the process of identifying groups of similar objects on the basis of a similarity measure so that each cluster contains objects similar to other objects within the same cluster and different from objects of other clusters [18]. Clustering on EDM can be an effective technique to group students based on their learning characteristics, individual learning style preferences, academic performance, and behavioral interaction [3]. Document clustering (also known as text clustering) is the process of employing clustering techniques on textual data to efficiently arrange, search, summarize, or retrieve text documents [4]. Document clustering can be employed to organize documents, systematic browsing of documents, corpus summarization, and document classification [2]. In data mining, there are many clustering techniques and algorithms. To process text documents with conventional clustering algorithms, text corpus has been generally represented as a feature matrix, either obtained based on term frequency, term presence, or term frequency-inverse document frequency (TF-IDF) weighting scheme. The conventional feature matrix-based text representation schemes suffer from high dimensionality and sparsity of feature vectors [2]. In response, topic modeling-based representations or neural language models can be utilized to effectively represent text corpora [34]. Neural language models provide dense representation of text documents with semantic properties with less manual preprocessing. Recently, neural language models, such as word2vec, fastText, and global vectors (Glove), have yielded promising predictive performances on several natural language processing tasks, such as sentiment analysis and topic extraction [29,53]. Recent empirical analysis on topic modeling indicates that the utilization of neural language models in conjunction with clustering methods can outperform the conventional clustering schemes [32].

Discussion forum posts can be utilized in different types of learning, including blended learning and MOOCs. Traditional place-based classroom environments and blended learning courses involve the physical presence of instructors and students. For such learning environments, informal discussions and meetings among the students and student–instructor communication can enhance the learning process of students. For MOOCs, discussion forum posts serve as the main channel to share opinions, analysis, responses, and feedback to the queries of others. Consequently, MOOC forum posts are more active, comprehensive, and rich source of information regarding the learning process, confusion, social interactions, and concerns, compared to the online forums in traditional courses [51]. To enhance learning on MOOCs, instructors should provide prompt responses to the queries of students. As the number of students enrolled in MOOCs can be high, the automated prioritization of question topics can be an essential task. In this regard, a document-clustering model on MOOC discussion forum posts based on weighted word-embedding schemes and clustering methods has been presented to identify question topics on discussion posts. Four word-embedding schemes (namely, word2vec, fastText, Glove, and Doc2vec), four weighting functions (i.e., TF-IDF, IDF, smoothed IDF, and subsampling function [SF]), and four clustering algorithms (i.e., K-means, K-means++, self-organizing maps [SOMs], and divisive analysis clustering [DIANA] algorithm) for document clustering and topic modeling on MOOC discussion forum posts have been evaluated. In total, 20 different feature representations have been obtained from word-embedding schemes and weighting functions. The feature representation schemes have been evaluated in conjunction with four clustering methods. For the evaluation task, the empirical results for the latent Dirichlet allocation (LDA) have been also included. The empirical results in terms of adjusted rand index (ARI), normalized mutual information (NMI), and adjusted mutual information (AMI) indicate that weighted word-embedding schemes combined with clustering algorithms outperform the conventional schemes. The purpose of this study is to obtain an efficient document-clustering model to identify question topics in MOOC discussion forum posts. To the best of our knowledge, this is the first comprehensive analysis on MOOC discussion forum posts, in which the performance of conventional clustering algorithms, word-embedding schemes, and weighting functions has been reported. The empirical results seek to answer the following research questions.

1. Can neural language models yield promising models for MOOC discussion forum posts?
2. Is there a statistically meaningful difference between the clustering quality obtained by conventional clustering algorithms represented as feature matrix and clustering schemes integrated by word-embedding models?
3. Which word-embedding scheme yields the highest performance on MOOC discussion forum posts?
4. Can weighting functions enhance the performance of neural language models for MOOC discussion forum posts?
5. Which clustering algorithm performs the best for identifying question topics on MOOC discussion forum posts?

The structure of this paper is as follows. In Section 2, the current research contributions on EDM have been reviewed. In Section 3, the methodology of the research has been presented, including a description to text corpus, word-embedding models, weighting function, the LDA, and clustering algorithms. In Section 4, the empirical results of the study have been presented with a discussion. Finally, Section 5 presents the concluding remarks.

2 | RELATED WORK

Natural language processing techniques have been employed to address problems on education field. Several applications include the identification of self-learning behavior, self-assessment, and students' learning evaluation [3]. In this section, the earlier works on the field have been presented, with a special emphasis on MOOCs.

Adamopoulos [1] utilized text mining and machine-learning methods on user-generated online reviews about MOOCs to model factors on student retention, such as course, platform, and university. In another study, Wen et al [50] employed sentiment analysis on MOOC discussion forum posts to identify students' trending opinions towards the course, lecture, and peer-assessment. The sentiment analysis indicated that there is a correlation between the sentiment ratio on daily forum posts and dropout rates of students. Sra and Chakraborty [43] conducted a survey to identify the opinion of instructors and undergraduate students from the computer science department on MOOCs. The survey results indicated that instructors and students view MOOCs as a comprehensive source of knowledge to advance the learning of students.

In another study, Ramesh et al [39] presented a topic modeling-based approach on MOOC discussion forum posts, where seeded LDA has been employed to uncover useful information to enhance student retention on MOOCs. Altrabsheh et al [5] also employed sentiment analysis on text feedbacks to identify the learning-related emotions of students. In this study, student feedbacks, opinions, and feelings about different courses have been collected from Twitter. To process text corpus, different N-gram models (i.e., unigram, bigram, and trigram) and their ensemble combinations were considered. Several machine-learning classifiers (i.e., Naïve Bayes, support vector machines, maximum entropy, and random forest classifier) have been evaluated in conjunction with different N-gram models.

Modeling student-learning habits by classifying their MOOC forum posts into four classes based on the ICAP framework, Wang et al [47] examined how different cognitive activities affected their learning benefit. Similarly, Wang et al [48] presented a topic modeling-based approach to identify whether engaging in higher order thinking behaviors results in more learning, compared to the general or focused attention to lecture materials. Another unsupervised modeling approach to understand MOOC discussion forums has been introduced in [15]. In this scheme, bag-of-words-based representation has been utilized to represent text corpus and the k-medoids clustering algorithm has been employed to model the data. In a similar way, Lee [26] utilized clustering to model the problem-solving patters in MOOCs. In this scheme, SOMs and hierarchical clustering method have been employed on log files of MOOCs to identify similar problem-solving patterns among the students with similar characteristics.

Recently, deep-learning-based approaches have been also employed for EDM tasks. For instance, Wei et al [49] introduced a transfer-learning-based model for MOOC discussion forum posts classification. In this scheme, an ensemble deep learning framework, which integrates a convolutional neural network and a long short-term memory, has been utilized to identify the urgency and sentiment orientation of forum posts. Similarly, Bustillos et al [12] examined the predictive performance of machine learning methods and deep learning architectures for sentiment analysis on an intelligent learning environment. In another study, Sun et al [45] presented a deep-learning-based scheme to identify urgent posts in MOOC discussion forums, involving immediate attention from the instructors. In this contribution, the predictive performance of conventional classifiers (such as Naïve Bayes, support vector machines, and random forest) and conventional deep learning models (such as convolutional neural network, recurrent neural network, long short-term memory, and gated recurrent unit) has been evaluated on MOOC discussion forum posts. Lin et al [27] examined the predictive performance of lexicon-based and machine learning-based schemes for opinion mining on student evaluations of teaching. Recently, Onan [33] examined the predictive performance of machine learning models and deep learning architectures for opinion mining on students' evaluation of teaching.

3 | METHODOLOGY

This section presents the methodology of our research, namely descriptions to text corpus, clustering algorithms, word-embedding schemes, weighting functions, and the LDA has been presented.

3.1 | Corpus

To collect a text corpus on MOOC discussion forum posts, five computer science courses provided by edX platform have been utilized. The courses are "Artificial Intelligence" course (ColumbiaX: CSMM.101x), "Machine Learning Fundamentals" course (UCSanDiegoX: DSE220x), "Big Data in Education" course (TeachersCollegeX—BDE1x), "Nature in Code: Biology in JavaScript" course (EPFLx—NiC1.0×), and "Agile Development Using Ruby on Rails—The Basics" course (BerkeleyX—CS169.1x). In this way, 935 discussion threads with 3,262 forum posts have been obtained. The forum posts were initially annotated manually, as either question or non-question. The question posts have been further annotated as follows: course content question (CQ), technique question (TQ), and course logistic question (LQ). The non-question forum post that is related to questions has been further annotated as follows: response of course content question (C), response of technical question (T), and the response of logistic question (L) [51]. Table 1 presents the distribution of posts in each category and Table 2 presents sample MOOC discussion forum posts from the corpus.

To annotate the raw MOOC discussion forum posts, an annotation process has been employed, where each post has been assigned either to question or non-question category. The posts of question category have been further annotated to three aforementioned categories. Two experts have annotated the raw discussion forum posts. Cohen's kappa (κ) metric has been computed. For the corpus, a κ of 0.83 has been achieved, which indicated a perfect agreement among the annotators [33,44].

To process text documents by conventional clustering algorithms, the basic preprocessing tasks have been employed on the corpus. For preprocessing stages, the basic scheme outlined in [49] has been adopted. Initially, text normalization has been employed. All letters in the text corpus have been converted to lowercase letters in this stage. In addition, all sequence and punctuation marks have been eliminated. In addition, all sequence and punctuation marks have been eliminated. Abbreviations have been converted into their expanded versions. In addition, uninformative parts of text documents (such as URLs, stop words, irrelevant words, and sparse terms)

TABLE 1 The distribution of posts in each category

Category	Number of forum posts
Course content question (CQ)	362
Response of CQ (C)	627
Technique question (TQ)	443
Response of TQ (T)	995
Course logistic question (LQ)	181
Response of LQ (L)	275

TABLE 2 Sample MOOC discussion forum posts

Discussion forum post	Category
For people using the CountVectorizer, are you setting ngrams_range = (1,2) for the bigram assignment or (2,2)? (2,2) seems to be right based on the phrasing of the assignment but I'm getting terrible scores for both bigrams sections. My unigram scores are nearly perfect though and that seems to be the only discriminating factor in the code.	Technique Question (TQ)
Aargh. I'm glad I came here and saw this thread before I wasted any more time banging my head on the bigrams! I think the phrasing of the assignment, as you say, clearly specifies (2,2). So it's simultaneously annoying, and a relief, to see my score go up 40 points with just the change to (1,2). It's just a hunch, and I'm probably totally wrong about this... but something tells me that whoever crafted this problem may have possibly copy-pasted code from "6.2.3.3. Common Vectorizer usage" here without modifying it: https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction	Response of TQ (T)
Hello, During this lecture´s video (close to the end) there is a piece of the lecture that is repeated two times (Estimating probabilities and Text Classification) and at least one slide (the second part of the Text Classification section) that is being skipped. Is this intended or a problem in the video edition? Thanks	Course Content Question (CQ)
Hi, You're right that the editing seems a bit weird, but the content looks correct. In the Naive Bayes video, she just goes back and forth through slides and repeats one question, but the subsequent material seems correct. She skips the second slide of text classification, but the material for it is covered in the next slide. Do you feel you missed some explanations? What do other students think?	Response of CQ (C)
Hello, I have a couple of questions about the previous steps for the final exam. 1) I´m going to use a laptop for the exam and I have a second screen as well as a keyboard and the Internet router on the desk. Should I remove everything from the desk? 2) About the room...I have a small library in the room I plan to do the exam. It is not close to the desk and of course, is not in front of me. Should I move out of the rooms any book? 3) About the ID. Should It be the same one that I used to enter in the verified track? I did it with my passport in that case but the lighting in the room and the plastic of the passport make it glitter and is easier to get the picture to my national ID. Is it permitted? Maybe look like stupid questions but I don´t want to fail the exam due to this kind of small detail.	Course Logistic Question(LQ)
Here is the video link regarding the instruction of final exam setting: https://support.edx.org/hc/en-us/articles/360034119314-Proctored-Exam-Quick-Start-Guide Please also do the practice proctored exam in advance! Good luck,	Response of LQ (L)

have been eliminated. The tokenization, that is, the process of separation of given sentences and words of documents into tokens and characters, has been performed. In addition, stemming on the text corpus has been performed to identify word stems. For the stemming task, the Snowball stemming algorithm has been utilized. All the preprocessing stages on the corpus have been performed on NLTK [28,38].

3.2 | Clustering algorithms

Clustering is an unsupervised task in machine learning in which data instances have been assigned to groups based on similarity. In the empirical analysis, four clustering algorithms (i.e., K-means, K-means++, SOMs, and DIANA algorithm) have been considered. The rest of this section briefly explains the algorithms.

- K-means algorithm (KM) is a partition-based clustering algorithm, which is frequently employed in the cluster analysis due to its easy implementation, simplicity, and efficiency [19]. KM takes the number of clusters as the input parameter. It is frequently employed in cluster analysis, due to its easy implementation, simplicity, and efficiency. The algorithm initiates with the random selection of *k* objects as cluster centers. The remaining objects are assigned to the clusters with closest centers. The algorithm continues to compute the new mean of clusters until stopping criterion. The algorithm gives good results especially for clusters with well-aligned and compact shapes [18]. It is an efficient and scalable algorithm.
- The performance of the K-means algorithm is greatly influenced by the initialization of random seeds that are selected as the cluster centers. K-means++ algorithm (KM++) utilizes a heuristic function to find the

initial cluster centers of the K-means algorithm [7]. The algorithm initiates with the identification of one cluster center uniformly at random from the data points. Then, the distances between data points and their nearest centers are computed. The cluster center is updated based on a weighted probability distribution. In this way, k cluster centers are selected. Afterward, the clustering process continues with the application of the conventional k-means algorithm.

- SOM is a partition-based clustering algorithm, which is based on clustering and dimension reduction [24]. SOM algorithm can be utilized in a wide range of applications, including sampling, supervised learning, and cluster analysis [46]. In this scheme, a two-dimensional grid with nodes is obtained. This grid functions as cluster centers in the high-dimensional space. Inner nodes of the grid are connected by an edge. For the data vectors, the nearest neighbors are determined by the learning algorithm. The SOM algorithm presents a visual representation for clustering results in a lower dimensional space, which enhances the interpretability of clustering results [16].
- Hierarchical clustering algorithms can be either agglomerative or divisive. In the agglomerative clustering algorithms, the individual objects are merged into clusters based on similarities, whereas the objects within a single group are divided into subgroups in divisive clustering algorithms. DIANA is a divisive hierarchical clustering algorithm [13]. The algorithm starts with an initial cluster containing all the objects. Then, a cluster with the maximum diameter is selected and the splinter group is initialized. For each data objects, an average distance to all other objects is computed. Based on the computed value, data objects are assigned into the splinter groups with the maximum difference [18].

3.3 | Word-embedding schemes

The conventional representation scheme utilized in text classification and document clustering is to represent text corpus with the use of a feature matrix obtained based on term-frequency, term-presence, or TF-IDF schemes. The conventional text representation schemes cannot capture the semantic relationships among the components of text documents. They also suffer from high dimensionality and sparsity of feature vectors [2]. Recently, neural language models have been successfully employed on natural language processing tasks [29]. Neural language models provide distributed learning representations of words in low-dimensional spaces [8]. Neural language models are based on the distributional hypothesis. According to these models, words with similar meanings are to be found in similar contexts. Neural language models aim to capture the similarity between words. Neural language models provide dense representation of text documents with semantic properties with less manual preprocessing. Dense vector representation is appropriate for many tasks in natural language processing, including document clustering [30,41]. In the empirical analysis, four word-embedding schemes (namely word2vec, fastText, Glove, and Doc2vec) have been considered. The rest of this section briefly explains neural language models utilized in the empirical analysis.

- The word2vec model, which is based on representing words as continuous vectors, is one of the most basic word-embedding models [11]. The basic idea of the Word2vec model is that multiple similarity relationships can be captured with continuous vectors during model training. Word2vec is an artificial neural network-based language modeling method that includes the input layer, the output layer, and the hidden layer. Word2vec includes two basic algorithms for training word vectors: continuous bag of words (CBOW) and skip-gram (SG). In the CBOW model, the context vector is obtained through the sum of word vectors in the context window. In the SG algorithm, the words surrounding the target word are determined over the target word. The CBOW algorithm can work effectively with a small amount of data. SG architecture is more costly in terms of computation and gives results that are more effective in large datasets [21].
- The fastText model is another artificial neural network-based language modeling method based on the word2vec model [52]. In this model, text representation is made over a bag of character *n*-grams instead of words. To create word vectors, subword information has been utilized. The fastText model is a computationally efficient representation model. The model offers a suitable representation structure for languages with many compounds. The fastText method offers a more effective word-embedding representation for rare words.
- The GloVe model is another unsupervised neural language model to effectively learn word embeddings from text documents. The model combines local context-based learning of the word2vec model with the global matrix factorization [37]. In the model, the probability ratios of words are taken into account for calculating the error function.
- The Doc2Vec model is an unsupervised neural language model to obtain vector-based representation for sentences, paragraphs, and documents [6]. Unlike the word2vec model, the Doc2Vec model provides a feature vector with an additional context for each

document contained in the text corpus. This document vector is trained along with word vectors.

3.4 | Weighting functions

In the conventional neural language models, such as word2vec, each word in the sentence has been assigned the equivalent weights and the word embeddings have been obtained by taking the mean of word embeddings. Recent empirical analysis on natural language processing tasks indicates that weighted averaging of word embeddings can improve clustering or classification performance [17]. In the empirical analysis, four weighting functions (i.e., TF-IDF, IDF, smoothed IDF, and SF) have been considered. The rest of this section briefly explains the weighting functions utilized in the empirical analysis.

IDF is one of the most commonly utilized weighting schemes in information retrieval. Let D denote the total number of documents in text corpus and D_i denote the total number of documents in which word i has been encountered. IDF has been computed based on the following equation [6]:

$$\mathrm{idf}(i) = \left(\frac{D}{D_i}\right). \tag{1}$$

Smoothed IDF (SIF) is an improved weighting function to assign weight values to terms in text documents [17]. Let n_i denote the total number of times word i encountered in the document and N the total number of terms encountered in the text corpus. TF (tf_i) for a particular word i has been computed based on (2). Taking term frequencies into account, smoothed IDF has been computed based on (3), where a is a parameter, which has been generally set to 10^{-4} and tf_{ic} denotes the corpus wide term frequencies [6]:

$$\mathrm{tf}_i = \left(\frac{n_i}{N}\right), \tag{2}$$

$$w_i = \left(\frac{a}{a + \mathrm{tf}_{ic}}\right). \tag{3}$$

SF is another weighting scheme, which aims to subsample frequent words. SF-based weighting has been computed based on the following equation:

$$w_i = \left\{\begin{matrix} \sqrt{\dfrac{t}{\mathrm{tf}_{ic}}} & \text{if} & \mathrm{tf}_{ic} \geq t \\ 1.0 & \text{if} & \mathrm{tf}_{ic} < t \end{matrix}\right\}, \tag{4}$$

where t is a parameter, which has been generally set to 10^{-5} and tf_{ic} denotes the corpus wide term frequencies.

TF-IDF is another common weighting scheme based on term frequency and inverse document frequency. TF represents the relative frequency of a word t in a text document and inverse document frequency scales with the number of documents in which a word has been encountered. Based on TF (tf_i, as computed by 2) and inverse document frequency (idf, as computed by 1), TF-IDF has been computed based on the following equation:

$$\mathrm{TF} - \mathrm{IDF} = \mathrm{tf}_i * \mathrm{idf}(i). \tag{5}$$

3.5 | Latent Dirichlet allocation

LDA is a generative topic-modeling model that assumes that text was generated from a discrete distribution of phrases known as topics, which together form a document that contains a probabilistic distribution of topics [10]. LDA seeks to define the underlying structure of the latent topic based on the information observed. In LDA, each document's words are the information observed. The words are produced in a two-stage operation for each document in the corpus. First, a topic distribution is randomly selected. Based on this distribution, a topic for each word of the document is randomly selected from the distribution over topics [9,35].

4 | EXPERIMENTS AND RESULTS

In this section, measures are utilized to evaluate document-clustering methods and experimental procedure and the experimental results have been presented.

4.1 | Evaluation measures

For evaluating document-clustering methods, three evaluation measures (namely NMI, ARI, and AMI) have been utilized.

Mutual information (MI) is a measure of identifying the mutual dependency of the two independent random variables. Let X and Y denote two discrete random variables with a joint probability distribution $p(x, y)$; the mutual information $MI(X, Y)$ has been computed as

$$\mathrm{MI}(X, Y) = \sum_{y \in Y} \sum_{x \in X} \log\left(\frac{p(x, y)}{p(x)p(y)}\right). \tag{6}$$

NMI is a normalization of MI to take values between zero (representing no mutual information) and one (representing perfect correlation). NMI has been computed by (7), where $H(X)$ and $H(Y)$ denote the entropies as given by (8):

$$\mathrm{NMI}(X, Y) = \frac{\mathrm{MI}(X, Y)}{\sqrt{H(X)H(Y)}}, \tag{7}$$

$$H(X) = -\sum\nolimits_{i=1}^{n} p(x_i)\log(p(x_i)) \tag{8}$$

ARI is an external cluster validation measure calculated as

$$\mathrm{ARI} = \frac{2(ad - bc)}{(a + b)(b + d) + (a + c)(c + d)}, \tag{9}$$

where a denotes the number of objects that are assigned to the same cluster in V and U, b denotes the number of pairs in the same cluster in U, but not in V, c denotes the number of pairs in the same clusters in V, but not in U, and d denotes the number of pairs at different clusters in U and V. For ARI, the higher value of it indicates the better clustering quality.

AMI is an adjustment of the mutual information, which has been computed as

$$\mathrm{AMI}(X, Y) = \frac{\mathrm{MI}(X, Y) - E\{\mathrm{MI}(X, Y)\}}{\max\{H(X), H(Y)\} - E\{\mathrm{MI}(X, Y)\}}, \tag{10}$$

where $E\{\mathrm{MI}(X, Y)\}$ denotes the expected value of the mutual information.

4.2 | Experimental procedure

In the empirical analysis, the clustering quality of word-embedding schemes and clustering algorithms on MOOC discussion forum posts has been evaluated. Four word-embedding schemes (i.e., word2vec, fastText, GloVe, and Doc2vec) have been evaluated to represent text corpus. Four weighting functions (i.e., TF-IDF, inverse document frequency, smoothed IDF, and SF) and four clustering algorithms (i.e., K-means, K-means++, self-organizing maps, and DIANA algorithm) have been considered. By combining word-embedding schemes by different weight functions, 20 different feature representations have been obtained based on neural language models. The feature representation schemes have been evaluated in conjunction with four clustering methods. For evaluation task, empirical results for the LDA and three conventional text representation schemes (i.e., term-frequency, term-presence, and TF-IDF-based representation) have been also included. For conventional text representation schemes, the matrix has been limited to the top 1,000 terms per document by frequency.

For neural language model-based representation, our own model has been trained to obtain word and document embedding. For word2vec, fastText, and Doc2vec, continuous SG and CBOW methods with varying vector sizes (vector size of 200 and 300) and different dimensions for projection layers (dimension size of 100 and 200) have been evaluated. As the highest performances for neural language models have been achieved for the vector size of 200 and dimension of projection layer of 200 with SG model, the results for these configurations have been listed in the empirical results. Word-embedding-based neural language models have been implemented on Tensorflow and *pyclustering* package has been utilized to implement the clustering algorithms. Unless otherwise stated, the default parameters for the algorithms on the packages have been utilized. To implement LDA-based topic modeling, gensim 3.4.0 package with the default parameter set has been employed.

TABLE 3 Evaluation measure values for conventional text representation schemes and clustering methods

Normalized mutual information				
Feature representation	K-means	K-means++	SOM	DIANA
TF	0.503	0.522	0.531	0.535
TP	0.492	0.517	0.528	0.534
TF-IDF	0.506	0.523	0.532	0.536
LDA	0.513	0.526	0.543	0.546
Adjusted rand index				
Feature representation	K-means	K-means++	SOM	DIANA
TF	0.501	0.526	0.535	0.538
TP	0.488	0.520	0.531	0.523
TF-IDF	0.506	0.528	0.537	0.539
LDA	0.511	0.549	0.552	0.560
Adjusted mutual information				
Feature representation	K-means	K-means++	SOM	DIANA
TF	0.506	0.521	0.528	0.539
TP	0.500	0.519	0.530	0.538
TF-IDF	0.509	0.522	0.531	0.541
LDA	0.516	0.526	0.536	0.546

Abbreviations: DIANA, divisive analysis clustering; LDA, latent Dirichlet allocation; SOM, self-organizing map; TF-IDF, term frequency-inverse document frequency.

4.3 | Experimental results

In this section, ARI, NMI, and AMI values obtained by conventional text representation schemes, clustering algorithms, and word-embedding schemes have been presented.

Table 3 presents evaluation measure values obtained by conventional text representation schemes in conjunction with four clustering algorithms. As it can be observed from the empirical results listed in Table 3, the lowest clustering qualities have been obtained by the term presence-based representation of text corpus. The second lowest performance has been achieved by TF-based representation. The highest performance among the conventional feature representation schemes has been achieved by the LDA scheme, which is followed by the TF-IDF weighting scheme.

In Table 3, the clustering qualities of four conventional feature representation schemes have been evaluated in conjunction with four clustering algorithms (i.e., k-means, k-means++, self-organizing maps, and DIANA algorithm). The empirical results indicate that the DIANA algorithm generally outperforms the other conventional clustering algorithms. The average cluster quality values obtained by self-organizing maps algorithm are generally higher than the results obtained by the k-means and k-means++ algorithm. The empirical results in terms of NMI, ARI, and AMI indicate that the highest clustering quality has been obtained by the utilization of LDA in conjunction with the DIANA algorithm. In general, the LDA-based topic modeling outperforms conventional text representation schemes.

In Tables 4–6, NMI, ARI, and AMI values obtained by word-embedding schemes in conjunction with weighting functions and clustering algorithms have been presented, respectively.

TABLE 4 Normalized mutual information (NMI) values for word-embedding schemes

Word embedding	Weighting functions	K-means	K-means++	SOM	DIANA
word2vec	Unweighted	0.546	0.590	0.552	0.564
fastText	Unweighted	0.521	0.519	0.532	0.546
GloVe	Unweighted	0.532	0.584	0.537	0.559
Doc2vec	Unweighted	0.566	0.597	0.586	0.611
word2vec	IDF weighted	0.677	0.669	0.667	0.728
fastText	IDF weighted	0.662	0.664	0.640	0.716
GloVe	IDF weighted	0.673	0.667	0.644	0.727
Doc2vec	IDF weighted	0.703	0.682	0.684	0.733
word2vec	SIF weighted	0.643	0.642	0.632	0.664
fastText	SIF weighted	0.613	0.632	0.628	0.637
GloVe	SIF weighted	0.619	0.636	0.631	0.657
Doc2vec	SIF weighted	0.655	0.649	0.638	0.685
word2vec	SF weighted	0.587	0.609	0.604	0.627
fastText	SF weighted	0.570	0.602	0.598	0.621
GloVe	SF weighted	0.585	0.603	0.601	0.625
Doc2vec	SF weighted	0.609	0.618	0.619	0.633
word2vec	TF-IDF weighted	0.747	0.721	0.809	0.782
fastText	TF-IDF weighted	0.713	0.689	0.694	0.745
GloVe	TF-IDF weighted	0.732	0.705	0.749	0.771
Doc2vec	TF-IDF weighted	0.788	0.735	0.812	**0.827**

Note: Bold values indicate the highest predictive performances for each configuration.

Abbreviations: DIANA, divisive analysis clustering; GloVe, global vectors; SF, subsampling function; SIF, smoothed inverse frequency; SOM, self-organizing map; TF-IDF, term frequency-inverse document frequency.

TABLE 5 Adjusted rand index (ARI) values for word-embedding schemes

Word embedding	Weighting functions	K-means	K-means++	SOM	DIANA
word2vec	Unweighted	0.543	0.589	0.571	0.591
fastText	Unweighted	0.520	0.583	0.543	0.548
GloVe	Unweighted	0.526	0.585	0.558	0.586
Doc2vec	Unweighted	0.588	0.594	0.615	0.604
word2vec	IDF weighted	0.693	0.692	0.726	0.745
fastText	IDF weighted	0.683	0.665	0.695	0.723
GloVe	IDF weighted	0.690	0.675	0.719	0.736
Doc2vec	IDF weighted	0.704	0.704	0.738	0.772
word2vec	SIF weighted	0.659	0.655	0.675	0.704
fastText	SIF weighted	0.654	0.642	0.664	0.692
GloVe	SIF weighted	0.657	0.652	0.669	0.692
Doc2vec	SIF weighted	0.670	0.659	0.675	0.723
word2vec	SF weighted	0.636	0.638	0.632	0.675
fastText	SF weighted	0.615	0.603	0.623	0.610
GloVe	SF weighted	0.627	0.616	0.626	0.621
Doc2vec	SF weighted	0.641	0.638	0.636	0.687
word2vec	TF-IDF weighted	0.747	0.797	0.765	0.789
fastText	TF-IDF weighted	0.728	0.752	0.749	0.781
GloVe	TF-IDF weighted	0.733	0.776	0.764	0.787
Doc2vec	TF-IDF weighted	0.772	0.801	0.785	**0.816**

Note: Bold values indicate the highest predictive performances for each configuration.
Abbreviations: DIANA, divisive analysis clustering; GloVe, global vectors; SF, subsampling function; SIF, smoothed inverse frequency; SOM, self-organizing map; TF-IDF, term frequency-inverse document frequency.

In Table 4, the empirical results in terms of NMI for neural language model-based representations have been presented in conjunction with four conventional clustering algorithms. It is clear from the table that weighting functions enhance the performance of neural language models for MOOC discussion forum posts. Regarding the performance of different word-embedding schemes, the Doc2vec model outperformed all other word-embedding schemes. The second highest clustering quality has been generally achieved by the word2vec model. The third highest clustering quality has been achieved by the GloVe model, and the lowest clustering quality has been achieved by fastText-based representation. As it can be observed from the empirical results presented in Tables 3 and 4, the clustering qualities obtained by word-embedding schemes outperform the clustering qualities obtained by conventional clustering algorithms represented as feature matrix. Regarding the clustering performance of different weighting schemes, four configurations (namely unweighted scheme, IDF-weighted scheme, SIF-weighted scheme, SF-weighted scheme, and TF-IDF weighted scheme) have been evaluated. As it can be observed from the empirical results presented in Tables 4–6, the TF-IDF weighting scheme outperforms the other weighting functions, which is followed by the IDF weighting scheme. The third highest clustering quality has been obtained by smoothed inverse frequency (SIF)-based weighting function, and the lowest clustering quality has been obtained by SF. The empirical results indicate that the DIANA algorithm outperforms the other conventional clustering algorithms. The second highest clustering qualities have been generally achieved by the self-organizing maps algorithm when word-embedding schemes have been utilized to represent text corpus. Among all the configurations taken into consideration in the empirical analysis, the highest clustering quality in terms of all evaluation measures has been achieved by Doc2vec-based representation in conjunction with the DIANA clustering algorithm, with an NMI score of 0.827, an ARI value of 0.816, and an AMI value of 0.868.

TABLE 6 Adjusted mutual information (AMI) values for word-embedding schemes

Word embedding	Weighting functions	K-means	K-means++	SOM	DIANA
word2vec	Unweighted	0.603	0.616	0.586	0.619
fastText	Unweighted	0.572	0.568	0.576	0.549
GloVe	Unweighted	0.593	0.609	0.583	0.595
Doc2vec	Unweighted	0.618	0.617	0.630	0.631
word2vec	IDF weighted	0.710	0.724	0.739	0.765
fastText	IDF weighted	0.680	0.671	0.712	0.744
GloVe	IDF weighted	0.690	0.716	0.728	0.756
Doc2vec	IDF weighted	0.729	0.744	0.742	0.767
word2vec	SIF weighted	0.668	0.665	0.708	0.739
fastText	SIF weighted	0.665	0.651	0.655	0.734
GloVe	SIF weighted	0.666	0.652	0.684	0.735
Doc2vec	SIF weighted	0.673	0.671	0.710	0.742
word2vec	SF weighted	0.640	0.643	0.639	0.683
fastText	SF weighted	0.629	0.634	0.633	0.633
GloVe	SF weighted	0.638	0.639	0.637	0.637
Doc2vec	SF weighted	0.661	0.643	0.640	0.728
word2vec	TF-IDF weighted	0.792	0.806	0.794	0.864
fastText	TF-IDF weighted	0.730	0.745	0.754	0.769
GloVe	TF-IDF weighted	0.744	0.762	0.773	0.815
Doc2vec	TF-IDF weighted	0.800	0.814	0.850	**0.868**

Note: Bold values indicate the highest predictive performances for each configuration.

Abbreviations: DIANA, divisive analysis clustering; GloVe, global vectors; SF, subsampling function; SIF, smoothed inverse frequency; SOM, self-organizing map; TF-IDF, term frequency-inverse document frequency.

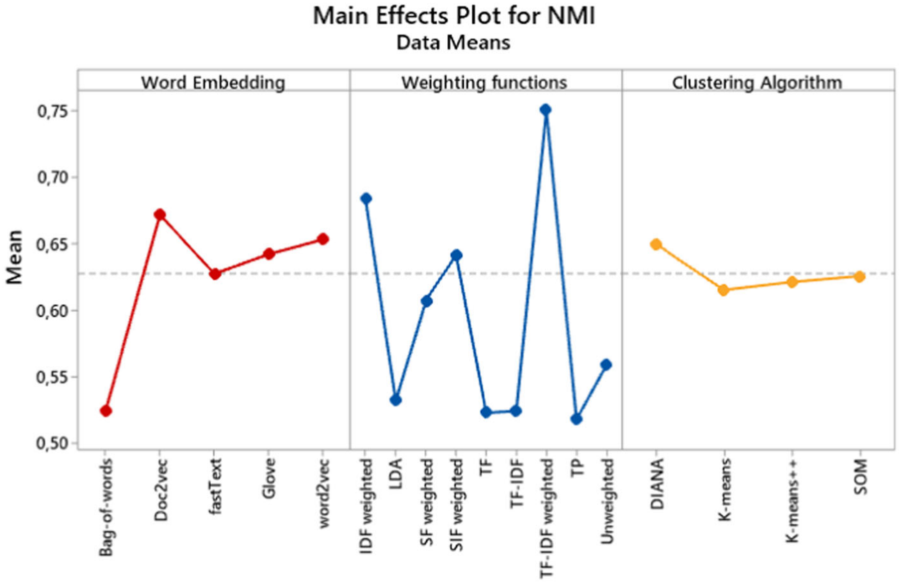


FIGURE 1 Main effects plot for normalized mutual information (NMI) scores. DIANA, divisive analysis clustering; GloVe, global vectors; LDA, latent Dirichlet allocation; SF, subsampling function; SIF, smoothed inverse frequency; SOM, self-organizing map; TF-IDF, term frequency-inverse document frequency

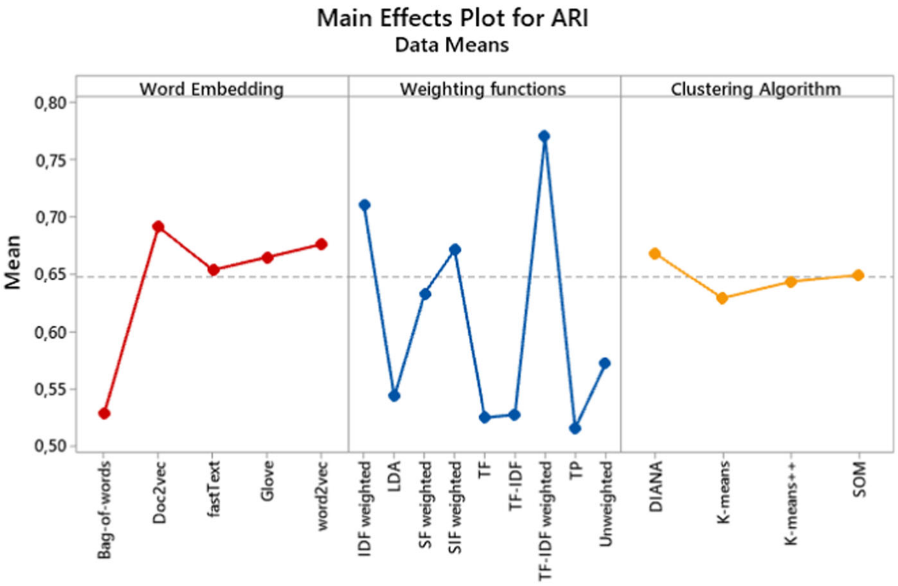


FIGURE 2 Main effects plot for adjusted rand index (ARI) scores. DIANA, divisive analysis clustering; GloVe, global vectors; LDA, latent Dirichlet allocation; SF, subsampling function; SIF, smoothed inverse frequency; SOM, self-organizing map; TF-IDF, term frequency-inverse document frequency

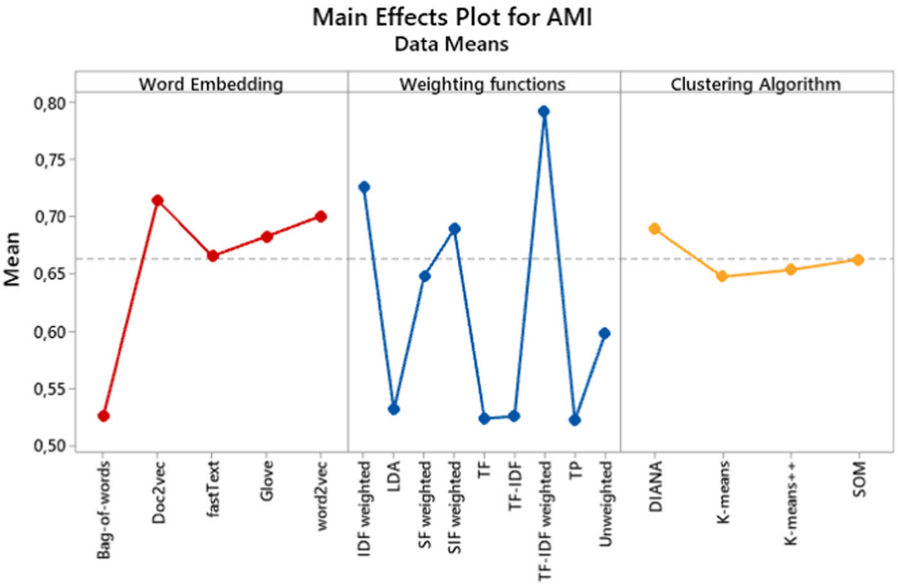


FIGURE 3 Main effects plot for adjusted mutual information (AMI) scores. DIANA, divisive analysis clustering; GloVe, global vectors; LDA, latent Dirichlet allocation; SF, subsampling function; SIF, smoothed inverse frequency; SOM, self-organizing map; TF-IDF, term frequency-inverse document frequency

In Figures 1–3, the main effects plots for NMI, ARI, and AMI scores have been presented to summarize the main findings of the empirical results.

To further evaluate the statistical significance of the empirical results, one-way analysis of variance (ANOVA) tests have been performed in the Minitab statistical program. In Table 7, the statistical significance of the results listed in the empirical analysis have been evaluated, where DF, SS, MS, *F*, and *p*, denote degrees of freedom, adjusted sum of squares, adjusted mean square, *F* value, and *p* value, respectively. According to the one-way ANOVA test results presented in Table 7, there is a statistically meaningful difference between the results obtained by different

TABLE 7 One-way ANOVA test results

Source	DF	Adj SS	Adj MS	*F* value	*p*
Feature representation	3	0.02686	0.008954	25.35	.000
Weighting functions	4	0.35421	0.088553	250.72	.000
Clustering algorithms	3	0.02299	0.007664	21.70	.000
Error	69	0.02437	0.000353		
Total	79	0.42844			

Abbreviations: Adj SS, adjusted sum of squares; Adj MS, adjusted mean square; ANOVA, analysis of variance; DF, degrees of freedom.

feature representation schemes, weighting functions, and clustering algorithms ($p < .0001$).

5 | CONCLUSION

MOOCs are proliferating quickly. For many MOOCs, however, learning efficiency and student retention are not satisfactory. One essential stage to enhance learning efficiency and student retention is to improve the interaction of instructors and students on MOOCs. The discussion forums serve as an essential channel for students, in which the opinions have been exchanged. MOOC discussion forum posts include text-based opinions and analysis, images and videos, reading posts, responses, and feedbacks to the queries of others. In traditional educational settings, face-to-face informal discussions among students and student–instructor communication also constitute an important part of the learning process. All these functionalities have been replaced by discussion forums on MOOCs. Hence, MOOC forum posts provide a rich source of information, regarding the students' learning processes, social interactions, and concerns. To enhance learning on MOOCs, instructors should provide prompt responses to the queries of students. In addition, misleading, inaccurate, or false answers provided to the students, as a response by the other students should be properly identified. The number of students enrolled in MOOCs can be high, which may degrade instructors' capacity to provide prompt responses to queries on course topics. Hence, machine learning-based schemes to identify question topics become a key issue on MOOC platforms.

In this paper, a deep learning model based on weighted word embedding and clustering has been introduced to identify question topics on MOOC discussion forums. The predictive performance of conventional text representation schemes, clustering algorithms, neural language models, and weighting functions have been evaluated. The empirical analysis on a corpus with 935 discussion threads with 3,262 forum posts indicates that the Doc2vec model in conjunction with TF-IDF weighted mean of representation in conjunction with DIANA clustering algorithm yields the highest clustering quality in terms of all compared evaluation metrics. The empirical analysis indicates that deep learning can be utilized to identify question topics and to prioritize discussion forum posts.

There are several practical implications for future research. As mentioned in advance, discussion forum posts are essential channels on MOOCs. However, students may spend less time on discussion forums due to the unsatisfactory responsiveness of the platforms. To fill this gap, artificial intelligence-based teaching assistants and question answering systems can be deployed in MOOC platforms. This can reduce the teaching loads of instructors and teaching assistants while enhancing the interactivity. As mentioned in advance, the immense quantity of questions has been shared by many students on MOOC platforms. Among these questions, providing students with topics that match their interests is a difficult task. Hence, machine-learning- and deep-learning-based question recommendation can be integrated to MOOC platforms.

ORCID

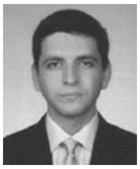
Aytuğ Onan http://orcid.org/0000-0002-9434-5880
Mansur Alp Toçoğlu http://orcid.org/0000-0003-1784-9003

REFERENCES

1. P. Adamopoulos. What makes a great MOOC? An interdisciplinary analysis of online course student retention. Proc. 34th Internat. Conf. Inf. Syst., 2013, pp. 1–21.
2. C. C. Aggarwal, and C. X. Zhai, *A survey of text clustering algorithms*, Mining text data (C. C. Aggarwal, and C. X. Zhai, eds.), Springer-Verlag, Berlin, Germany, 2012, pp. 77–128.
3. H. Aldowah, H. Al-Samarraie, and W. M. Fauzy, *Educational data mining and learning analytics for 21st century higher education: A review and synthesis*, Telematics Inform. **39** (2019), no. 4, 13–49.
4. L. AlSumait, and C. Domeniconi, *Text clustering with local semantic kernels*, Survey of text mining II (M. W. Berry, and M. Castellanos, eds.), Springer-Verlag, Berlin, Germany, 2008, pp. 87–105.
5. N. Altrabsheh, M. Cocea, and S. Fallahkhair. *Sentiment analysis: Towards a tool for analysing real-time students feedback*. IEEE 26th Internat. Conf. Tools With Artificial Intelligence, 2014, pp. 419–423.
6. S. Arora, Y. Liang, and T. Ma, *A simple but tough-to-beat baseline for sentence embeddings*, Proc. Internat. Conf. Learn. Representations, 2017, pp.1–4.
7. D Arthur and S Vassilvitskii, *k-means++: The advantage of careful seeding*, Proc. 18th Annual ACM-SIAM Symp. Discrete Algorithms, 2007, pp.1027–1035.
8. Y. Bengio et al., *A neural probabilistic language model*, J. Mach. Learn. Res. **3** (2003), no. Feb, 1137–1155.
9. D. M. Blei, *Probabilistic topic models*, Commun. ACM **55** (2012), no. 4, 77–84.
10. D. M. Blei, A. Y. Ng, and M. I. Jordan, *Latent Dirichlet allocation*, J. Mach. Learn. Res. **3** (2003), no. Jan, 993–1022.
11. P. Bojanowski et al., *Enriching word vectors with subword information*, Trans. Assoc. Comput. Linguistics **5** (2017), 135–146.
12. R. Bustillos et al., *Opinion mining and emotion recognition in an intelligent learning environment*, Comput. Appl. Eng. Educ. **27** (2019), no. 1, 90–101.
13. H. Chipman and R. Tibshirani, *Hybrid hierarchical clustering with applications to microarray data*, Biostatistics **7** (2005), no. 2, 286–301.
14. S. Donitsa-Schmidt and B. Topaz, *Massive open online courses as a knowledge base for teachers*, J. Educ. Teach. **44** (2018), no. 5, 608–620.
15. A. Ezen-Can et al. *Unsupervised modeling for understanding MOOC discussion forums: A learning analytics. approach*. Proc. 5th Internat. Conf. Learn. Anal. Knowledge, 2015, pp. 146–150.

16. E Glaab, Analysing functional genomics data using novel ensemble, consensus and data fusion techniques, Ph.D. dissertation, Dept. Comput. Sci., Nottingham Univ., Nottingham, UK, 2011.
17. V. Gupta et al. (2018). Unsupervised document representation using partition word-vectors averaging. In The International Conference on Learning Representations (ICLR) 2019, pp. 1–28.
18. J. Han and M. Kamber, Data mining: Concepts and techniques, 2nd ed., Morgan Kaufmann, San Francisco, CA, 2006.
19. A. K. Jain, *Data clustering: 50 years beyond k-means*, Pattern Recognit. Lett. **31** (2010), no. 8, 651–666.
20. M. Jia et al., *Who can benefit more from massive open online courses? A prospective cohort study*, Nurse Educ. Today **76** (2019), 96–102.
21. A. Joulin et al., FastText. zip: Compressing text classification models, arXiv:1612.03651, 2016.
22. A. M. Kaplan and M. Haenlein, *Higher education and the digital revolution: About MOOCs, SPOCs, social media, and the Cookie Monster*, Bus. Horiz. **59** (2016), no. 4, 441–450.
23. I. U. Khan et al., *Predicting the acceptance of MOOCs in a developing country: Application of task-technology fit model, social motivation, and self-determination theory*, Telematics Inform. **35** (2018), no. 4, 964–978.
24. T. Kohonen, Self-organizing maps, Springer, Berlin, Germany, 2001.
25. V. Kovanović et al., *Exploring communities of inquiry in massive open online courses*, Computers & Education **119** (2018), 44–58.
26. Y. Lee, *Using self-organizing map and clustering to investigate problem-solving patterns in the massive open online course: an exploratory study*, Journal of Educational Computing Research **57** (2019), no. 2, 471–490.
27. Q. Lin et al., *Lexical based automated teaching evaluation via students' short reviews*, Comput. Appl. Eng. Educ. **27** (2019), no. 1, 194–205.
28. E. Loper and S. Bird. *NLTK: The natural language toolkit*, Proc. ACL-02 Workshop Effective Tools Methodol Teach. Natural Lang. Process. Comput. Linguistics, 2002, pp. 63–70. https://doi.org/10.3115/1118108.1118117.
29. T. Mikolov et al., Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, 2019, pp. 3111–3119, arXiv:1310.4546.
30. T. Mikolov et al., Efficient estimation of word representations in vector space, 2013, arXiv:1301.3781.
31. P. J. Muñoz-Merino et al., *Precise effectiveness strategy for analyzing the effectiveness of students with educational resources and activities in MOOCs*, Comput. Human. Behav. **47** (2015), 108–118.
32. A. Onan, *Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering*, IEEE Access **7** (2019), 145614–145633.
33. A. Onan, *Mining opinions from instructor evaluation reviews: A deep learning approach*, Comput. Appl. Eng. Educ. **28** (2020), 117–138.
34. A. Onan, H. Bulut, and S. Korukoglu, *An improved ant algorithm with LDA-based representation for text document clustering*, J. Inf. Sci. **43** (2017), no. 2, 275–292.
35. A. Onan, S. Korukoglu, and H. Bulut, *LDA-based topic modelling in text sentiment classification: An empirical analysis*, Int. J. Comput. Linguistics Appl. **7** (2016), no. 1, 101–119.
36. A. Peña-Ayala, *Educational data mining: A survey and a data mining-based analysis of recent works*, Expert Syst. With Appl. **41** (2014), no. 4, 1432–1462.
37. J. Pennington, R. Socher, and C. Manning. *GloVe: Global vectors for word representation*. Proc. Conf. Empirical Methods Natur. Lang. Process., 2014, pp. 1532–1543.
38. M. F. Porter. *Snowball: A language for stemming algorithms*, 2001. http://snowball.tartarus.org/.
39. A. Ramesh et al. *Understanding MOOC discussion forums using seeded LDA*. Proc. 9th Workshop on Innov. Use NLP Building Educ. Appl., 2014, pp. 28–33.
40. C. Romero and S. Ventura, *Educational data mining: A survey from 1995 to 2005*, Expert Syst. Appl. **33** (2007), no. 1, 135–146.
41. C. W. Schmidt. Improving a tf-idf weighted document vector embedding, 2019, arXiv:1902.09875.
42. G. Siemens and R. S. Baker. *Learning analytics and educational data mining: Towards communication and collaboration*. Proc. 2nd Internat. Conf. Learn. Anal. Knowledge, 2012, pp. 252–254.
43. P. Sra and P. Chakraborty, *Opinion of computer science instructors and students on MOOCs in an Indian university*, J. Educ. Technol. Syst. **47** (2018), no. 2, 205–212.
44. E. Stamatatos, *A survey of modern authorship attribution methods*, J. Am. Soc. Inf. Sci. Technol. **60** (2009), no. 3, 548–556.
45. X. Sun et al. *Identification of urgent posts in MOOC discussion forums using an improved RCNN*. IEEE World Conf Eng. Educ. (EDUNINE), 2019, pp. 1–5.
46. V. Vesanto and E. Alhoniemi, *Clustering of the self-organizing map*, IEEE Trans. Neural Netw. Learn. Syst. **11** (2000), no. 3, 586–600.
47. X. Wang et al., Investigating how student's cognitive behavior in MOOC discussion forums affect learning gains, Paper presented at the Internat. Conf. Educational Data Mining, 2015, pp. 226–233.
48. X. Wang, M. Wen, and C. P. Rosé. *Towards triggering higher-order thinking behaviors in MOOCs*, Proc. 6th Internat. Conf. Learn. Anal. Knowledge, 2016, pp. 398–407.
49. X. Wei et al., *A convolution-LSTM-based deep neural network for cross-domain MOOC forum post classification*, Information **8** (2017), no. 3, 92.
50. M. Wen, D. Yang, and C. Rose. Sentiment analysis in MOOC discussion forums: What does it tell us? Proc. Educ Data Mining, 2014, pp. 130–137.
51. Y. Xu and C. F. Lynch (2019). What do you want? Applying deep learning models to detect question topics in MOOC forum posts? In 2019 KDD workshop on Deep Learning for Education (DL4Ed), pp. 1–6.
52. L. Zhang, S. Wang, and B. Liu, *Deep learning for sentiment analysis: A survey*, Data Mining Knowledge Discovery **8** (2018), no. 4, 848–870.
53. Y. Zhang et al., *Does deep learning help topic extraction? A Kernel K-means clustering method with word embedding*, J. Informetrics **12** (2018), no. 4, 1099–1117.

AUTHOR BIOGRAPHIES



Aytuğ Onan received the B.S. degree in computer engineering from the Izmir University of Economics, İzmir, Turkey, in 2010, and the M.S. degree in computer engineering and the Ph.D. degree in computer engineering from Ege University, Turkey, in 2013 and 2016, respectively. He has been an Associate Professor with the Department of Computer Engineering, Izmir Katip Celebi University, Izmir, Turkey, since April 2019. He has published several journal articles on machine learning and computational linguistics. Dr. Onan has been an editor for the KSII Transactions on Internet and Information Systems and an Associate Editor for the Journal of King Saud University Computer and Information Sciences.

Mansur Alp Tocoglu received the B.Sc. degree in software engineering and the M.Sc. degree in artificial intelligent systems from the Izmir University of Economics, İzmir, Turkey, in 2008 and 2013, respectively, and the Ph.D. degree in computer engineering from Dokuz Eylül University, İzmir, Turkey. He has been working as a Research Assistant with the Software Engineering Department, Manisa Celal Bayar University, Manisa, Turkey. His research interest includes information extraction from text using machine learning techniques.

How to cite this article: Onan A, Toçoğlu MA. Weighted word embeddings and clustering-based identification of question topics in MOOC discussion forum posts. *Comput Appl Eng Educ*. 2021;29:675–689. https://doi.org/10.1002/cae.22252