Information Processing and Management 59 (2022) 103107

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Deep learning based conference program organization system from determining articles in session to scheduling

Esra Gündoğan, Mehmet Kaya *

Department of Computer Engineering, Firat University, Elazig, Turkey

A R T I C L E I N F O

Keywords:
Document similarity
Clustering
Scheduling
BERT
Organizing conference programs

A B S T R A C T

It is very important to create the conference programs correctly in terms of timing and content by preventing problems such as being of articles that do not have a common topic with each other in the same sessions, the parallel of the sessions containing articles on the same topic. It greatly affects the efficiency of conference for participants. Currently, conference programs are organized manually. Considering the conference scope and the number of articles in that conference, it is a difficult and time-consuming process. In this study, an automatic solution to this problem is presented. The use of the SBERT method is provided a more accurate calculation of article similarities compared to baseline methods and is increased the success of other stages. Unlike classical clustering methods, an approach that clusters in such a way that there are equal numbers of data points in the clusters is proposed. In order to find the topic of the clusters determined as sessions, a topic determination approach is proposed that takes into account both keyword and article content similarities. Furthermore, with the proposed approach for session scheduling, the conference program has been planned more effectively by considering the parallel sessions. The ICTAI conference has been chosen to test the proposed approach. The proposed program is compared with both the real program and the programs created using Word2vec and Glove methods. With the proposed program, 10% improvement is achieved in terms of session similarity. In addition, parallel sessions are better planned with no conflicts compared to the real program.

1. Introduction

In recent years, rapid development in science has been seen due to extensive research work in academia and industry. As a result of the development in science, researchers around the world publish research papers to share their scientific findings. These articles are very important for academic interaction (Fortunato et al., 2018; Khan et al., 2016). Conferences are one of the important platforms where articles are published for scientific results to reach researchers. Conferences bring researchers together to be aware of the latest studies and to produce new ideas. Since the publication time is shorter than the journals, conferences are largely preferred by researchers to publish articles. For example, in the Computing and Information Science publication category, it was reported in Scopus (2020) that while 62.3% of the articles published in Scopus were conference articles, 32.8% were journal articles. Hundreds of conferences on different topics are held every year in many parts of the world. Thousands of articles are accepted at these conferences. A good program is needed to present all the articles within certain days. It is currently prepared manually by the organizers.

* Corresponding author.
E-mail address: kaya@firat.edu.tr (M. Kaya).

https://doi.org/10.1016/j.ipm.2022.103107
Received 18 May 2022; Received in revised form 5 September 2022; Accepted 28 September 2022
Available online 12 October 2022
0306-4573/

Considering that software solutions are offered to many problems today, it is seen as a need to automate this problem, which is still solved manually.

In conferences, it is a time-consuming process to manually divide the articles into sessions, determine the titles of the sessions, and place each session at the appropriate timeslots in the conference timeline. In addition, as the number of articles increases, this process sometimes becomes unbearable. While organizing the sessions, the main purpose is to collect articles on a common topic in the same session and to enable participants listen to the sessions according to their expertise areas or interest. In some cases, articles that are not close to each other are brought together. Participants in such sessions leave the session after listening to one or two articles. This is undesirable in terms of conference effectiveness. In addition, when there is more than one session on the same topic in the conference program, it can be observed that these sessions are held in parallel. In this case, the participant cannot attend other sessions even though he/she is interested. However, participants should be given the opportunity to listen to as many presentations as possible. As a result, not well-organized sessions are not very efficient for the participants. In addition, the fact that the participants do not benefit from the conference effectively creates a bad impression for the conference owners. There may be a decrease in the number of participants in the conferences to be organized in the next years. Therefore, it is necessary to minimize these problems faced by both participants and conference owners and to create more effective and efficient conference programs.

1.1. Motivation

Papers accepted by the conference may belong to the same or different research fields. In the conference programs, the articles on the common topic are brought together to create sessions and these sessions are placed on the conference timeline. These sessions need to be created and scheduled correctly for conference efficiency. Considering the problems mentioned above for participants and conference owners, the aims of the study are as follows:

- Placing articles on the similar topics as possible in the same session
- Ensuring equal number of articles in each session
- Determining appropriate standard title for each session
- Preventing parallel scheduling of sessions on the same topic
- Ensuring participants to listen to as many presentations as possible
- Increasing the efficiency of conference programs

1.2. Contribution

Organizing conference program is a process that consists of many stages and different methods should be used. In order to perform this process automatically for the determined aims, the paper offers the following contributions:

- Proposing the use of SBERT for text representation has increased success compared to baseline methods.
- Unlike traditional clustering methods, a different clustering approach is proposed with as equal elements as possible in each cluster.
- An approach that takes into account article content similarity in addition to word distributions is proposed for topic modeling.
- A novel method is proposed for the session scheduling problem by assigning similar-themed sessions to different non-parallel timeslots.
- The difficult and time-consuming program preparation process has been simplified for the organizers.
- An efficient conference program has been automatically prepared for the participants and conference owners.
- To the best of our knowledge, this study is the first effort in solving the mentioned problem.

In this study, the process of preparing the conference program is carried out in four stages. In the first stage, the content similarities of the articles are used to create sessions. The title, abstract, keywords and references are used in the similarity calculation. The SBERT method is employed to find similarities. In the second stage, while placing the articles in the sessions, it is also taken into account to ensure that there are equal articles in each session, as well as the similarity scores between the articles. For this purpose, a clustering approach is proposed so that the number of articles is equal in each cluster (session). With this method, the sessions including equal number of articles are created based on the similarity scores of the articles. In this paper, automatic generation of ICTAI conference program, which accepts a large number of articles in the computer science field is given as illustrative example. In the third stage, session titles are created from computer science field. Considering the keyword and content similarities, appropriate titles sessions are determined. Finally, a session scheduling approach is proposed to distribute the sessions created in the last stage to the conference timeline. When the proposed program is compared with the real conference program, it is seen that the sessions in the proposed program consist of articles with higher similarity and an improvement is achieved in the real program. In addition, although several sessions on the same topic take place in parallel in the real program, it is corrected in the proposed program. Thus, a more effective conference program is created.

The rest of the study is organized as follows: Section 2 includes the literature review. Section 3 presents the proposed approaches for each stage. Section 4 gives detailed results for finding article similarities, determining sessions, assigning session titles and scheduling. Finally, Section 5 concludes the paper.

2. Literature review

Related works on each stage of the proposed method are examined. Document similarity approaches for the first stage, document clustering approaches for the second stage, keyword extraction approaches for the third stage and scheduling approaches for the last stage are separately presented.

2.1. Methods for document similarity

Document similarity is a measure of how close a document is to other documents, semantically and syntactically. In order to calculate this similarity, the documents must first be converted to numeric values. The best representation model of the documents is the Vector Space Model (Younge & Kuhn, 2016). One of the most frequently used methods is to calculate the similarities by modeling and weighting the documents in vector space. The similarity between documents can be calculated by dot product similarity, cosine similarity or a different similarity method of document vectors.

In the vector space model, documents are represented in a multidimensional vector space. The number of unique words in documents determines the size of the vector space. In this model, words are weighted according to their importance and documents can be represented as vectors by using these weight values. The Term Frequency - Inverse Document Frequency (TF-IDF) method (Qaiser & Ali, 2018), which considers both the frequency of the words and their effect within the whole cluster, is generally used in vector weighting. The Bag-of-Words (BOW) method (Steinert & Hoppe, 2016) is a document representation method that uses the fixed-size vector representation. However, this method has disadvantages and they have been eliminated by the deep learning-based Word2vec method, which has attracted great interest recently (Gong et al., 2019; Liu et al., 2017). One of the disadvantages, the size of the BOW representation vector is equal to vocabulary size. Document vectors are created by counting the co-occurrence statistics of words, resulting in sparse vectors. As the dimension of the vector increases, so does computational complexity. However, in the Word2vec method, the vector size is not as much as the number of unique words in the corpus. The size of the vector can be selected based on the corpus size and subject and thus dense vectors are created.

Other disadvantage, word order is not taken into account in vector space, so different sentences can have exactly the same representation when using the same words. However, the Word2vec method captures semantic information but does not use co-occurrence statistics. Glove (Pennington et al., 2014) calculates similarity by taking into account the co-occurrence rates of the words. The Doc2Vec method (Le & Mikolov, 2014) is also based on Word2Vec. Instead of just using words to predict the next word, another feature vector specific to the document has been added. It is a vector that remembers what is missing from the current content or acts as a memory holding the topic of the paragraph. It runs faster and requires less memory space as there is no need to store word vectors. Word2vec and Doc2vec methods are frequently used in text processing (Gündoğan & Kaya, 2019, 2020). One of the popular methods recently is the BERT (Devlin et al., 2018) method, which vectorizes words depending on the context. It eliminates the problem of word embedding models about generating identical vectors for synonyms.

2.2. Methods for document clustering

Manual tagging of data is difficult when the amount of data is very large. In addition, the information required for manual tagging is not always available or insufficient. Therefore, clustering is a more suitable method (Ravindran & Thanamani, 2015). Grouping documents according to their similarities is called document clustering. In document clustering, it is aimed to group documents according to their type or content. Many methods have been proposed in the literature for document clustering. Hierarchical and partitional clustering methods are the core methods (Afzali & Kumar, 2019).

The hierarchical clustering algorithm is based on the aggregation of similar features or vice versa division. There are two basic approaches: agglomerative and divisive. In the study by Cao et al. (2018), documents are clustered with the hierarchical clustering method using named entity features. In the study using the Birch (Balanced Iterative Reducing and Clustering using Hierarchies) hierarchical clustering approach (Kathiravan & Kalaiyarasi, 2015), the documents are clustered by looking at the semantically related sentences in the documents.

Partitional clustering takes the k parameter as input and divides a dataset consisting of n objects into k clusters. Clusters are created according to a specified criterion, such as minimizing the sum of squared error. The most important partitional clustering methods are K-means, Kmeans++ and K-medoids. In the study where the K-means method is used (Kongwudhikunakorn & Waiyamai, 2020), the words in the document are represented with the distributed word representation, the similarity between the documents is calculated with the word mover's distance and the short texts are clustered. In the hybrid study (Sarkar et al., 2014) where particle swarm optimization (PSO) and k-means algorithm are combined, PSO algorithm is used for initial centroid selection and then k-means algorithm is applied. This hybrid method is seen to be more successful than PSO and k-means algorithms in document clustering.

2.3. Methods for keyword extraction

Keyword extraction methods are grouped as statistical approach, linguistic approach, graph-based approach and machine learning approach.

In the statistical approach, keywords are extracted using statistical information such as word frequency, term frequency (TF), term frequency-inverse document frequency (TF-IDF) word co-occurrences, etc. In the study (Nyandag et al., 2017), in which keywords are extracted from a single document without the need for a corpus, the CHI-Square method, which calculates the degree of bias between a word and the most frequent words, and word frequency - inverted word frequency measure, which shows how important a word is from other words in the document, have been used.

The linguistic approach (Nguyen & Kan, 2007) uses the linguistic features of words to extract keywords. It is a language-dependent approach that includes syntactic analysis, discourse analysis, lexical analysis, etc.

The graph-based approach is based on finding important words or phrases in the created document graph. The document is defined as a graph in which the words are nodes and the relationships between words are edges. Edge relations; co-occurrence relations, syntax relations, semantic relations, etc. can be created on many principles (Beliga et al., 2015). TextRank (Zhang et al., 2020) is one of the graph-based approaches. It calculates the scores of the words in the document using PageRank and creates a graph. Initially, all the words in the network are candidate keywords. Then, by calculating the PageRank scores, the top-n words with the highest score are determined and given as keywords. In the study (Chen et al., 2019), which extracts keywords from the high-order structural features of the word co-occurrence graph, an approach that extracts keywords from the document itself without the need for a large corpus is proposed.

In the machine learning approach, keyword extraction is seen as a learning problem. For this purpose, training data and training models are needed. Keywords extracted from the training data are used to train the model. The model is then used to extract keywords from new documents. Hidden Markov model (Zhang et al., 2017), support vector machine (SVM) (Domoto et al., 2016), Naive Bayes (NB), etc. are employed as training models. In a supervised machine learning approach (Guleria et al., 2021), keywords based on statistical and linguistic features have been extracted using SVM. The performance of deep learning approaches in natural language processing has also popularized their use in keyword extraction. The use of convolutional neural networks (CNN), recurrent neural networks (RNN), long short term memory (LSTM) etc. models to extract keywords has provided very successful results. In the study

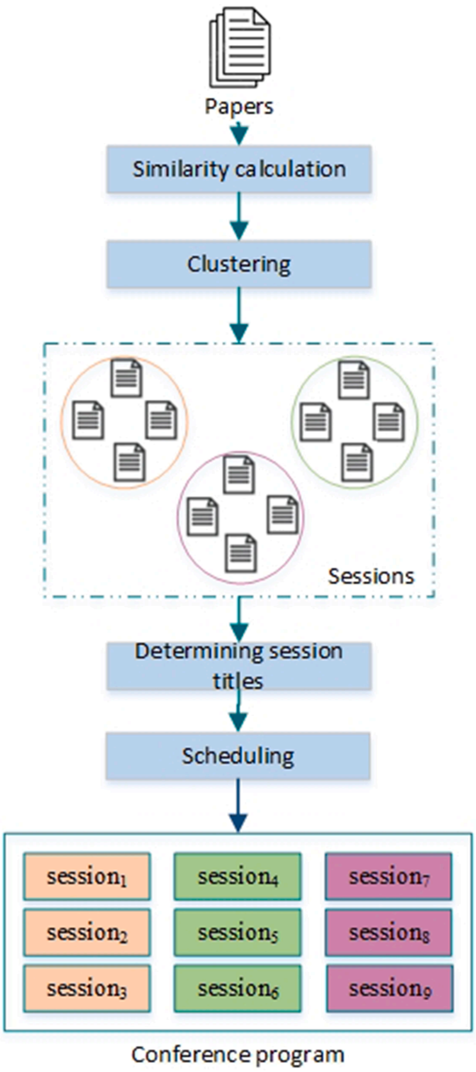


Fig. 1. Block diagram of organizing conference program.

where the RNN-based encoder-decoder model is proposed, key phrases are extracted from the scientific text (Meng et al., 2017). Wang & Zhang (2017) proposed a bi-directional long short-memory (LSTM) recurrent neural network (RNN)-based model to extract keywords from online product reviews.

2.4. Methods for scheduling

Scheduling is a problem that needs to be planned correctly in order to increase efficiency in many areas (Arunarani et al., 2019). Extensive research has been done on the scheduling problem in various fields. Different solutions have been presented to many problems with the genetic algorithm. In the study by Squires et al. (2022), it has been aimed to increase the operational efficiency of a medical center by scheduling medical appointments. In another study (Arık, 2022), a genetic algorithm has been used for permutation flow shop scheduling problems. Flight approach, departure sequencing problem and runway allocation problems are solved with terminal area flight scheduling by using a genetic algorithm (Liu et al., 2022). Another approach proposed for scheduling problems is graph coloring. In the study where graph coloring is used for cloud systems (De, 2022), the latency in waiting time is reduced and processes are executed better by correctly scheduling limited resources to multiple processes. In the study (Deogratias, 2022), in which graph coloring is used for the class timetable, a timetable has been created for teachers in which lesson collisions are reduced and productivity increases. In the proposed scheduling approach (Nandal et al., 2021) to create the college course timetable, optimal scheduling of lessons and classes for students and teachers has been provided, and a timetable without conflicts has been created.

3. Methodology

The conference program organization process is carried out in four stages. In the first stage, the similarities of the articles are calculated. In the second stage, clustering based on similarities is performed and sessions are created. In the third stage, appropriate titles for the sessions are determined. In the final stage, the sessions are placed on the conference timeline. The block diagram of this process is given in the Fig. 1.

3.1. Calculating of article similarity

The study is based on the similarities of the articles. Sessions are created according to the similarity of the articles with each other and other stages are carried out later. Therefore, similarity scores significantly affect the success of the study.

While finding similarities between documents, the whole document can be taken as input. When considering research articles, each part is written to give specific information. Therefore, same parts are more similar to each other. For example; abstract and reference are created in a different structure from each other. Abstract is a text that gives a general idea about the topic of the study. Reference, on the other hand, is a section that contains articles on a common topic with the article. While the similarity of abstract and reference parts is less, the similarity between abstract and abstract parts may be higher. Therefore, it is thought that finding the pairwise similarities of the article parts will give more correct results.

To better measure the semantic relations between article documents, we propose to use transformers-based BERT (Devlin et al., 2018) and its variant SBERT (Reimers & Gurevych, 2019). BERT is a state-of-the-art pre-trained and transformer-based language model that provides extremely high performance for similarity comparison between sentence pairs (Laskar et al., 2020; Yoo et al., 2021). It consists of several transformer encoder layers that provide feature extraction at both token-level and sentence-level. In order to improve the sentence-level computational performance of BERT, siamese network architecture is used in BERT. Siamese network consists of two artificial neural networks. Each sub neural network takes the data, maps it to a high-dimensional feature space, and outputs a representation of the data. The similarity of the two input sentences can be found by calculating the distance between the two representations. SBERT generates semantically meaningful sentence embeddings using siamese and triplet network structures. Unlike BERT, SBERT is fine-tuned on sentence pairs using a siamese architecture. It has two identical BERTs in parallel that share the same network weights. SBERT applies a pooling operation to the output of BERT and produces a fixed-size sentence embedding. The cosine distance between two sentence embeddings is calculated as in (1).

$$\text{Cosine Similarity} = \frac{V_{sentence_i} \cdot V_{sentence_k}}{\| V_{sentence_i} \| \cdot | V_{sentence_k} \|} \tag{1}$$

We used SBERT to produce each sentence embeddings in the document. The title+keyword, abstract and reference parts are given as separate inputs to the SBERT model. Different vector representations were obtained for the title+keyword, abstract and reference parts of each article. To calculate the similarity of the two articles, pairwise vectors of the same parts were used. The similarity between each pair was determined by calculating the cosine similarity between their embeddings. The final similarity score is determined by averaging the similarity scores calculated for the three parts of the two articles.

3.2. Creating sessions

Each session has equal time in the conference program. Therefore, it is important that the number of articles in the sessions is equal, the time is used correctly, the time allocated to the presentation of each session is the same, and the sessions are completed in the same time. To create the sessions, it is necessary to group the articles according to their similarity scores. Community discovery and clustering algorithms can be used for this purpose (Beniwal et al., 2020; Cozzolino & Ferraro, 2022). Communities are groups of nodes in a network that are strongly connected with each other and are weakly connected to nodes in the rest of the network. The nodes are grouped by extracting the communities from the created network. In the process of determining the communities, it is not possible to execute the algorithm so that there are equal nodes in each community. Therefore, the clustering approach is more suitable.

In clustering approaches, data points can be grouped by giving the number of clusters as a parameter or by bringing together similar items without knowing the number of clusters. Cluster sizes can be increased or decreased with various parameters. In this approach, the number of data in each cluster is used as a parameter. A useful approach is offered for applications where equal division is required, unlike other methods. In addition, methods initiated with different parameters such as cluster centroid selection may yield different clustering results. However, in this method, different clusters can occur if only two clusters have the same score. Because the distance is the same, different clusters are valid. This does not affect the success too much.

In this study, each cluster is determined as a session. Considering the conference time and the number of parallel sessions, the number of sessions (number of clusters) is certain. However, in a clustering based on the number of sessions, it is not guaranteed to have equal articles in each cluster. Therefore, a clustering that takes into account cluster size is required. Inspired by the agglomerative hierarchical clustering approach, a clustering method is proposed provided that each cluster contains equal data points. In agglomerative clustering, initially, all data points are a separate cluster. These clusters are brought together according to their similar features and a single cluster is obtained. There are many methods for calculating the distance between clusters. The most common are single, complete and average linkage. Apart from these methods; ward, weighted, centroid and median methods are also used (Murtagh & Contreras, 2017).

Single linkage calculates the minimum distance between two data points in different clusters. The closeness of the two clusters is measured by the minimum distance. Complete linkage calculates the maximum distance between two data points in different clusters. The closeness of the two clusters is measured by the maximum distance. Average linkage is found by calculating and averaging the distance between each pair of data points in two clusters. The closeness of the two clusters is measured by the average distance.

In agglomerative hierarchical clustering, two clusters with the closest distance are combined to create a new cluster. The relation between the data in the study is the similarity score. Therefore, in the proposed approach, the two clusters with the highest similarity score are merged. The closeness of the two clusters is defined as:

$$Sim(C_x, C_y) = \frac{1}{n_x n_y} \sum \{ sim(a_i, a_j) \mid a_i \in C_x,\ a_j \in C_y \} \tag{2}$$

where C_x and C_y are two clusters, a_i and a_j are data points in C_x and C_y, $sim(a_i, a_j)$ is similarity score between a_i and a_j data points, n_x and n_y are the number of data points in C_x and C_y.

Algorithm 1 shows the proposed clustering method. $N,\ S$ and *csize* represent the number of elements, similarity matrix and cluster size respectively. $C = \{C_1, C_2, \ldots,\ C_k\}$ are clusters created with the proposed algorithm. The algorithm runs as follows:

(1) Each iteration is run with non-clustered elements. Initially, all elements are added to the list. If the number of non-clustered elements is greater than the cluster size and at least one cluster is created as a result of the iteration, the iteration is repeated. At each iteration, clusters with *csize* are created. If no new cluster is created as a result of iterations, the remaining elements are considered as a cluster.
(2) The similarity matrix is updated to include only the similarity scores of the non-clustered elements. Elements clustered in this way are not included in clustering again, as clusters with maximum similarity are merged.
(3) The clusters with the highest similarity score are selected.
(4) It is checked whether the elements in the cluster are in the previously created clusters. If the element in the cluster is in a previously created cluster, other elements are added to that cluster. If it is not included in any cluster, it is determined as a new cluster. This process is repeated until every element is included in a cluster. The matrix is updated according to the average score of the clusters.
(5) The size of the clusters is checked and clusters with the pre-defined cluster size are added to final cluster list. It is removed from the iteration cluster list and these elements are not included in the clustering process. In this way, if clustering has occurred as a result of iteration, a new iteration is started. If there is no clustering, it is terminated.
(6) The remaining elements in the iteration cluster list are added to the list of non-clustered elements. A new iteration for clustering is started. If the non-clustered elements are less than the cluster size and no new clustering occurs as a result of iteration, the algorithm terminates.
(7) Finally, the final cluster list is returned.

The aim is to create clusters with as many pre-defined size as possible. However, this is not always possible. Some of the problems encountered during clustering and their solutions are as follows:

- If no new cluster is formed as a result of iteration, the algorithm is terminated and the remaining elements are considered as a cluster. This termination may occur in the first iterations and the number of non-clustered elements may be large. In this case, a new cluster is created with the most similar non-clustering elements to encourage clustering. The elements of this cluster are removed and the algorithm is run again. In this way, clustering is achieved again.

- The number of elements to be clustered may not be enough to distribute it equally to each cluster, and clusters smaller than the cluster size may be obtained as a result of clustering. In this case, the pre-defined number of clusters is taken into account. If the pre-defined number of clusters is not reached, the clusters smaller than the cluster size are merged. If the number of clusters has been reached and there are still non-clustered elements, these elements are included in the clusters with which they are most similar.

The proposed clustering approach creates clusters that contains equal elements. For this algorithm, the space complexity is $O(n^2)$, and the time complexity is $O(kn^3)$ where n represents the number of elements to be clustered, k denotes the number of clusters. Determining the highest similarity score, merging clusters, and updating the similarity matrix requires $O(n^3)$ computation. Since the number of elements to be clustered and the cluster size are certain, the number of clusters is calculated as k. It is repeated k times until the pre-defined cluster size is reached. The runtime of algorithm increases proportionally as the number of articles.

3.3. Topic determining

The title is one of the most important parts that give a general idea about the article's topic. Therefore, the titles can be used to determine the topic of the session. In addition, the references are also very informative about the article's topic. Studies on the used approach, method, data, etc. are included in the references. Articles and references belong to common topics. Therefore, the title and references are used for topic determining. The similarity of this parts for topic and session articles are calculated as in Section 3.1. Thus, the similarity of the sessions with each topic is determined.

In the proposed method, keyword similarities are taken into account as well as title and reference similarities. Keywords are one of the most important parts of the article. Therefore, the high similarity of session and topic keywords indicates that they may belong to a common topic. In this study, we employed KeyBERT (Zhang et al., 2019) algorithm to extract keywords and key phrases. KeyBERT is a deep learning model and uses BERT embeddings and cosine similarity to extract keywords. The words most similar to the document itself are determined as keywords. For this purpose, first, for document-level representation, the whole document is input to BERT and document embeddings are obtained. Second, for word-level representation, the words in the document are input to BERT and word embeddings are obtained. Finally, the cosine similarity between each word embedding and the document embedding is calculated. The words with the highest similarity score are determined as keywords. Words with a high similarity score are the words that will best represent the document.

The block diagram of the proposed topic determination method is given in the Fig. 2. First, the title and reference parts of session articles and topic articles are extracted and the similarity score of these parts is determined with SBERT. Then, the title and abstract parts of session articles and topic articles are extracted. With the KeyBERT method, the keywords belonging to the topic and the session are determined and the similarity of the keywords is found with BERT. By calculating the average of keyword similarity and content similarity, the similarity score of each topic with the session is determined. The topic with the highest similarity score is given as the title of the session.

3.4. Session scheduling

Once the papers in the sessions are determined, the task is to place the sessions on the conference timeline. Scheduling is very important for conference efficiency. Sessions belonging to the same topic should not be parallel while they are placed. In order to create the most efficient scheduling of the sessions, an approach has been proposed that considers the common sessions not to be placed in parallel.

There are several important criteria in the scheduling of conference programs: (i) number of days, (ii) number of time slots, (iii) number of sessions, (iv) number of parallel sessions, (v) sessions on the common topic

In the proposed approach, a room is defined for each session. The number of sessions is equal to the number of rooms. Since the sessions will be parallel, rooms are also parallel. This parallelism is taken into account when numbering rooms. An example room plan is given in the Table 1. The program is carried out in 2 days and 3 timeslots. If there are 2 parallel sessions in each timeslot, sequential numbers are given to the parallel sessions according to the timeslot, starting from the first day.

The basic idea in the proposed approach is that parallel rooms are not given to sessions with the same title. Therefore, matrices are defined that show both the relations between sessions and the relations between rooms. If the number of rooms is N, a room matrix

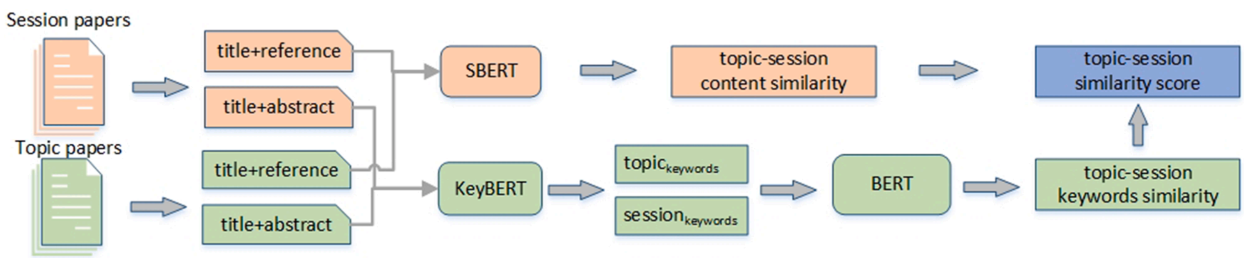


Fig. 2. Block diagram of topic determining.

Table 1
An example of room plan.

	Timeslot 1	Timeslot 2	Timeslot 3
Day 1	$Room_0$	$Room_2$	$Room_4$
	$Room_1$	$Room_3$	$Room_5$
Day 2	$Room_6$	$Room_8$	$Room_{10}$
	$Room_7$	$Room_9$	$Room_{11}$

R_{NxN} is created. The relation between rooms is defined if they take place in the same timeslot on the same day. If R_i and R_j are on the same day and in the same timeslot, $R[i][j] = 1$ in the room matrix. If the number of sessions is M, a session matrix S_{MxM} is created. The relation between sessions is defined if they have the same title. If S_i and S_j have the same title, $S[i][j] = 1$ in the session matrix. When a session will be placed in the room, the sessions it is connected to are accessed from the session matrix, and the rooms where that session cannot be placed are accessed from the room matrix.

The steps of the proposed approach are as follows:

Step (1) Session and room matrices are created.
Step (2) Sessions are sorted by their degree (the number of sessions they are connected to).
Step (3) The session with highest degree is taken.
Step (4) The sessions that this session is connected to are determined.
Step (5) If rooms are given to the sessions to which they are connected (parallel sessions), the rooms to which these rooms are connected are determined. These rooms are stated as rooms that will not be assigned to the session.
Step (6) One of the rooms that are empty and not determined as non-assignable to the session in step 5 is selected. The session is assigned to that room.
Step (7) The fullness status of the assigned room is set to true.
Step (8) Step 3 is repeated once the number of sessions. If all sessions are placed, the algorithm is terminated. If all sessions are not placed, the next session is selected as the start session and the algorithm is started again.

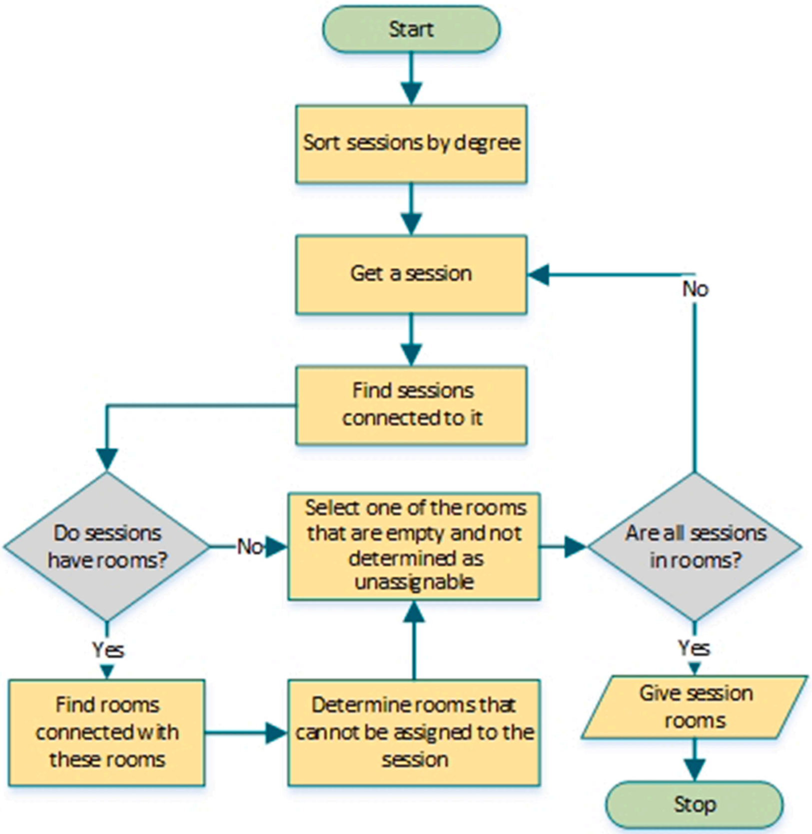


Fig. 3. Flow diagram of scheduling method.

The flowchart for the proposed scheduling method is given in the Fig. 3. Each session should not be in parallel with sessions with the same session title. Therefore, when a session is placed in a room, sessions with the same title are checked. If there is a session that it is connected to and a room is assigned to this session, no session can be assigned to any room to which this room is parallel. In this way, sessions with the same title are prevented from being assigned to parallel rooms. In some cases, with the selected start session, not all sessions can be placed in rooms. In this case, the next session is selected in the list of sessions sorted by their degree. With this session, the algorithm is restarted. As a result, a conference timeline with minimal parallelism is obtained.

In the proposed approach, sessions are distributed in rooms assigned for a certain day and time slot. For this algorithm, the space complexity is $O(n^2)$, and the time complexity is $O(kn^2)$ where n represents the number of sessions, k denotes the number of rooms. The runtime of algorithm increases proportionally as the number of sessions.

4. Experiments

The approach presented for organizing conference programs was carried out in 4 stages. The performance evaluation of each stage and the overall results are explained in detail below.

4.1. Dataset

First, the ICTAI (International Conference on Tools with Artificial Intelligence) 2021 conference program is selected for the implementation of the proposed method. ICTAI is a conference with a wide scope in the field of computer science that accepts articles on machine learning, big data, artificial intelligence, data mining, etc. topics. Then, we tested IEEE SMC (IEEE International Conference on Systems, Man and Cybernetics), which is wider scope than ICTAI, ICDABI (International Conference on Data Analytics for Business and Industry), which is narrower scope than ICTAI, and ASONAM (International Conference on Advances in Social Network Analysis and Mining), which includes relatively more specific topics. ICTAI, IEEE SMC, ICDABI and ASONAM 2021 conference information is given in Table 2. The characteristics of conference data used for experiments is summarized in Table 3.

The number of articles in each session in the conference program of ICTAI is mostly 7. However, since the total number of articles is not evenly distributed to the sessions, there are 8 articles in a few sessions. Therefore, the proposed program is planned to have 7 articles in each session. Since there are 3 articles increasing, the number of some sessions is 8 as in the real program.

It is one of the aims of the study that the session titles are standard topic titles. Therefore, 33 topics have been identified in the field of computer science. In order to determine the topic of the sessions, there is a need for articles that reflect the content of each topic. Therefore, a dataset containing articles for each topic is created. 100 articles for each topic are obtained from the Arxiv data repository.

4.2. Article similarity results

The high rate of pairwise similarity between the articles shows that those belong to a common topic. In order to show the performance of the proposed method for article similarity, 200 articles belonging to 10 different categories are selected from the dataset created for the topics. There are equal numbers of articles in each category.

Title, abstract, keywords and reference parts are used to determine article similarities. To examine the effect of these parts on similarity scores, the similarity of 200 test articles in different combinations is calculated with the SBERT method. In order to show the performance of the SBERT method, it has been compared with Word2vec and Glove, which are baseline methods used in document representation. Similarities are found between the articles in each category. The average similarity scores of the articles in the same category are given in the Fig. 4(a). Articles in the same category are expected to have high similarity rates. The results show that the Glove method gives higher similarity between articles in the same category compared to Word2vec. The SBERT model, on the other hand, has the highest similarity scores compared to the other two models. Title and keywords are the most important parts when looking at the effect of article parts on similarity. In general, the title, keywords and abstract parts are used to determine document similarities. However, references are just as important as the abstract. Looking at the results, adding the reference part has increased the similarity rate. For this reason, title, keyword, abstract and reference parts are used to determine article similarities.

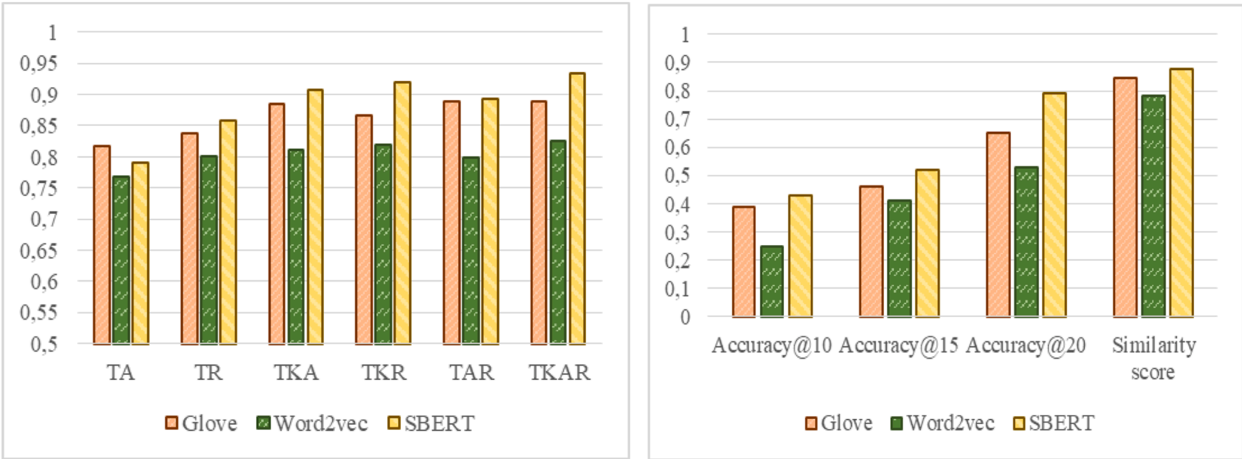
The fact that the articles in the same category have the highest similarity scores increases the probability of being included in the same clusters. Therefore, obtaining the highest similarity scores between the test article and articles in the same category directly affects the clustering performance in the next stage. In order to show whether the highest scores are obtained from the same category articles, the similarity ranking of the other articles to the test article is determined. With (3), the probabilities of whether the articles in

Table 2
ICTAI, IEEE SMC, ICDABI and ASONAM conference information.

	ICTAI	IEEE SMC	ICDABI	ASONAM
Number of articles	227	545	129	47
Number of sessions	32	113	27	12
Number of articles per session	7–8	4–6	4–5	3–4
Number of days	3	3	2	3
Number of time slots	3	4	3	2
Number of parallel sessions	4	10	10	2

Table 3
Summary of conference data.

	Num. of documents	Avg. document length	Num. of sentences	Avg. sentence length	Num. of words	Num. of unique words
ICTAI	227	1272	24,536	18	288,829	258,632
IEEE SMC	545	1303	38,265	22	452,378	402,356
ICDABI	129	1196	18,526	15	190,233	175,023
ASONAM	47	1025	4234	16	64,523	60,230



(a) **(b)**

Fig. 4. (a) Effect of article parts on similarity (T: Title R: Reference A: Abstract K: Keywords) (b) Similarity performance for papers in the same category.

the same category are included in the top-n ranking are determined. In the Fig. 4(b), as can be seen from the results obtained with the SBERT method, the number of articles in the same category is higher in the first-n ranking of the test article. The lowest results are obtained with Word2vec. This shows that it has better performance in SBERT similarity calculation compared to other methods.

$$Accuracy@N = \frac{number\ of\ papers\ in\ same\ category}{top - N\ papers} \tag{3}$$

4.3. Clustering results

The clearest indicator of the performance of the proposed clustering approach is to ensure that articles on the same topic are included in the same cluster. In order to measure the clustering performance, firstly, 200 articles selected from 10 different categories are clustered with 20 articles in each cluster. As a result, the percentages of articles from the same category in the obtained clusters show the success of the clustering. Table 4 shows the results of the clusters obtained. The topic that has more articles in the cluster is determined as the title of the cluster. According to the results, the similarity scores inner-cluster and the probability of 20 articles being included in the same cluster are quite high. This indicates that similar articles have been successfully grouped in the same cluster. For Information Retrieval and Software Engineering clusters, this rate is relatively lower. The reason for this is that these topics have close

Table 4
Clustering performance.

Cluster	Topic	Similarity score	Accuracy(%)
1	Artificial Intelligence	0.855	0.70
2	Machine Learning	0.814	0.75
3	Computer Vision and Pattern Recognition	0.901	0.90
4	Human-Computer Interaction	0.886	0.85
5	Information Retrieval	0.672	0.55
6	Multiagent Systems	0.767	0.70
7	Neural and Evolutionary Computing	0.703	0.65
8	Software Engineering	0.598	0.50
9	Networking and Internet Architecture	0.789	0.65
10	Cryptography and Security	0.865	0.80

relations with the topics belonging to other clusters. Due to the closeness of the topics in computer science, an article can be examined in more than one topic title. Therefore, articles belonging to these two clusters may have been included in other clusters. For this reason, it should not be evaluated as an incorrect clustering performance.

In the proposed clustering method, each cluster is determined as one session. The high similarity scores of the determined sessions show a correct clustering. As a result of clustering, 32 sessions are determined. A similarity score is obtained for each session by averaging the pairwise similarity scores of the articles in these sessions. Separate sessions are created according to the article similarity scores found with the SBERT, Glove and Word2vec methods. The similarity score comparison of the sessions created with three different methods is given in the Fig. 5(a). In general, sessions created based on the SBERT method have higher similarity scores. While the Word2vec session average similarity score is 0.76, the Glove session average similarity score is 0.81, the SBERT session average similarity score is 0.89. These scores show that more similar articles were collected in the same cluster in the clustering created by the SBERT method compared to Word2vec and Glove methods. Calculation of similarity at sentence level enabled article content similarities to be calculated more accurately.

The comparison of the similarity score of the created sessions with the real sessions is given in the Fig. 5(b). When both session clusters are examined, it is seen that the proposed method is at least as successful as the real program. While the average session similarity score for the real program is 80%, the average similarity score of the sessions determined by the proposed method is 90%. As a result, the articles are brought together more successfully with the proposed method. Thanks to the proposed method, the real program has been improved.

4.4. Session titles determination results

The topics of the sessions are determined by calculating keyword similarity and title+reference similarity. First, the keywords of the articles in the sessions and of the articles in the topics should be extracted. KeyBERT method is used for keyword extraction. In order to compare the performance of the KeyBERT method with the TextRank and YAKE algorithms, 100 test papers have been selected from the conferences in the dataset and the keywords of these papers have been extracted. By comparing the extracted keywords (kw_{method}) with the paper's keywords (kw_{paper}), precision, recall and F-measure criteria are calculated as in (4, 5 and 6). As seen in Table 5, the KeyBERT method has determined the keywords of the article more accurately compared to the TextRank and YAKE algorithms.

$$Precision = \frac{kw_{paper} \cap kw_{method}}{kw_{method}} \tag{4}$$

$$Recall = \frac{kw_{paper} \cap kw_{method}}{kw_{paper}} \tag{5}$$

$$F - measure = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \tag{6}$$

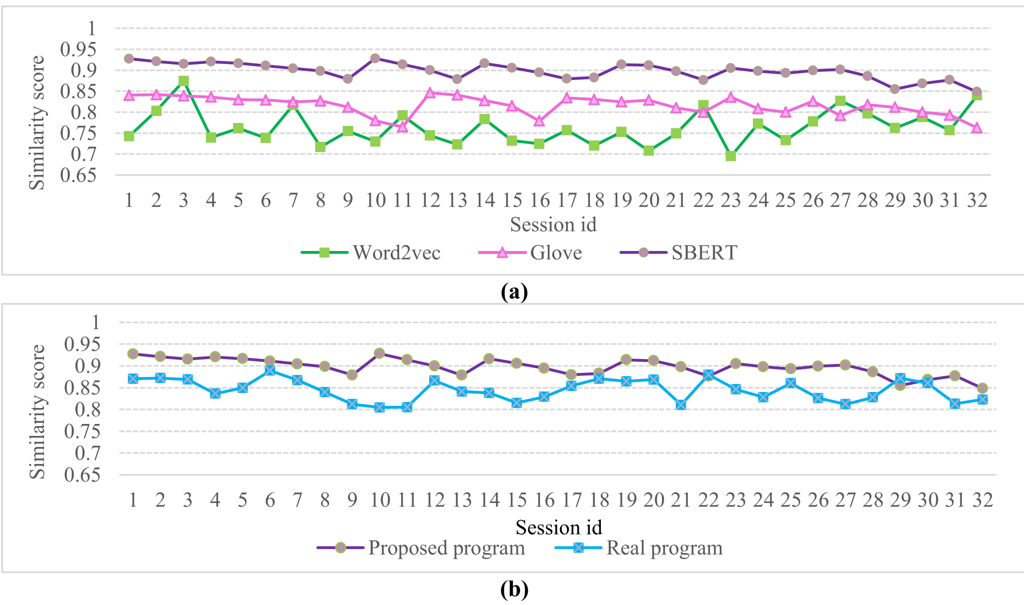


Fig. 5. (a) Comparison of similarity scores of sessions created with Word2vec, Glove and SBERT methods (b) Comparison of similarity scores of proposed and real program.

Table 5
Performance comparison of keyword extraction methods.

	Precision	Recall	F-measure
KeyBERT	0.75	0.68	0.71
TextRank	0.61	0.57	0.59
YAKE	0.70	0.65	0.67

A topic title can be easily determined from keywords. As an example, keywords belonging to the topic of Computer Vision and Pattern Recognition are extracted by using three methods. The extracted keywords and their scores are given in the Fig. 6. In general, common keywords are extracted. However, the keywords extracted with KeyBERT reflect the topic more clearly. In addition, the obtained scores with KeyBERT are higher than the other two methods. The effect of the BERT model on performance is also seen in keyword extraction.

In the Fig. 7, the keywords extracted by KeyBERT of the articles in a session whose session title is determined as Computer Vision and Pattern Recognition are given. Looking at the first 10 keywords, KeyBERT and the keywords extracted from the Computer Vision and Pattern Recognition topic are largely the same. The same keywords are obtained with high scores. Extraction of the right keywords has a great impact on finding similarities between topic and session articles. In this way, more successful assignment of topics to the sessions is achieved.

Topic modeling methods are used to determine which topic a collection of documents belongs to. In these methods, topic analysis is made by looking at the distribution of words in the documents. In the proposed approach, instead of just looking at the words, the similarities of the contents of the documents are also taken into account. In similar topics, looking only at keywords does not give accurate results. In general, keywords for these topics are common. Therefore, the content similarities of the documents are also used to determine the topic. The effect of keyword and content similarities on the success of topic determination is shown in the Fig. 8. For a sample session titled Computer Vision and Pattern Recognition, similarity scores of the top-10 titles are shown. These titles are Computer Vision and Pattern Recognition (CVPR), Neural and Evolutionary Computing (NE), Machine Learning (ML), Graphics (GR), Artificial Intelligence (AI), Human-Computer Interaction (HC), Mathematical Software (MS), Computational Geometry (CG), Robotics

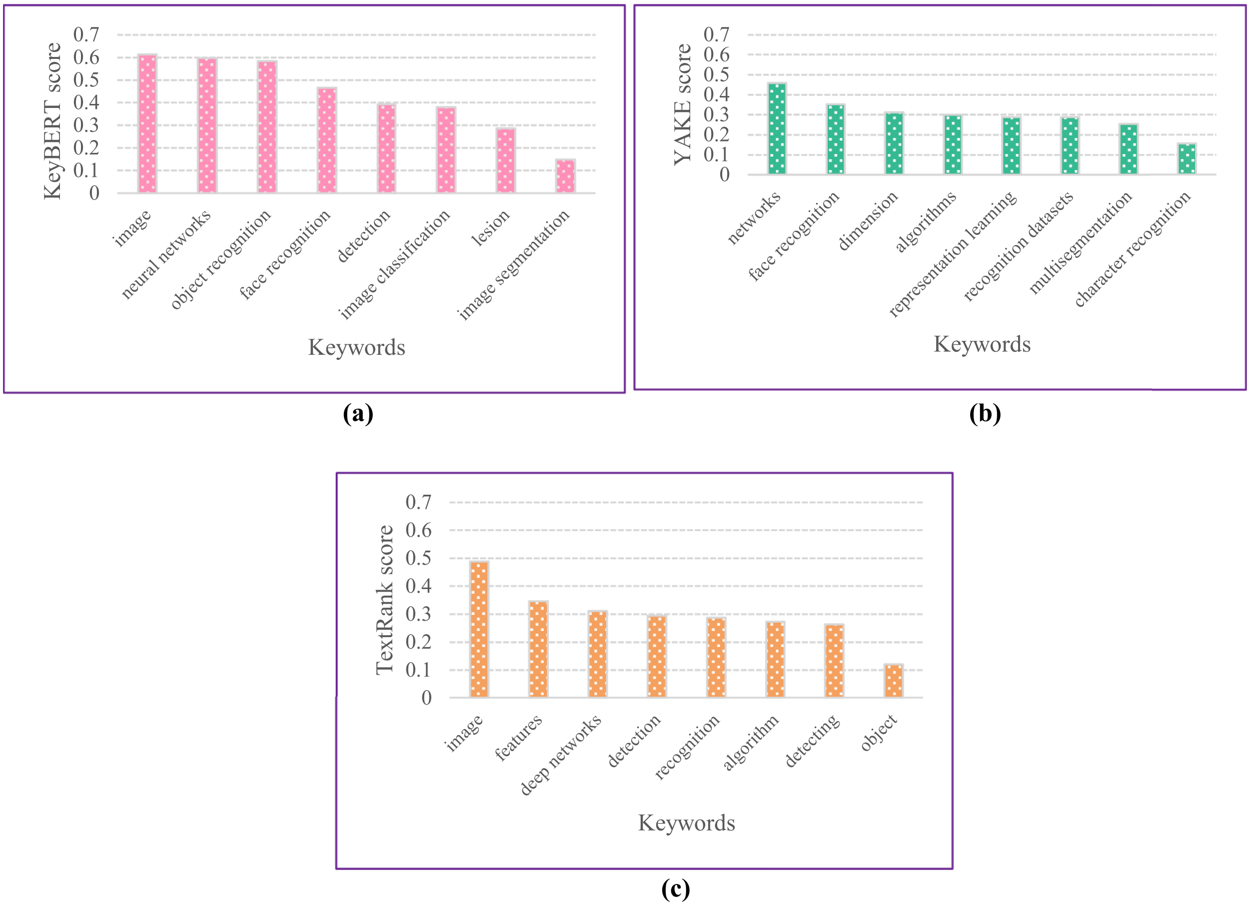


Fig. 6. Keywords and scores (a) KeyBERT (b) YAKE (c) TextRank.

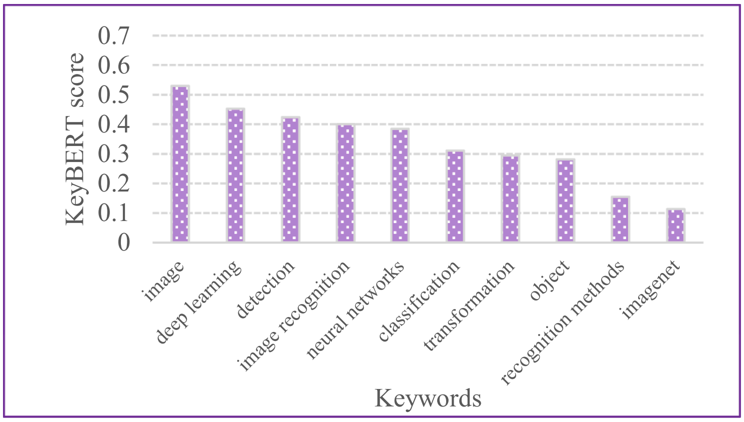


Fig. 7. Keywords and scores of CVPR session.

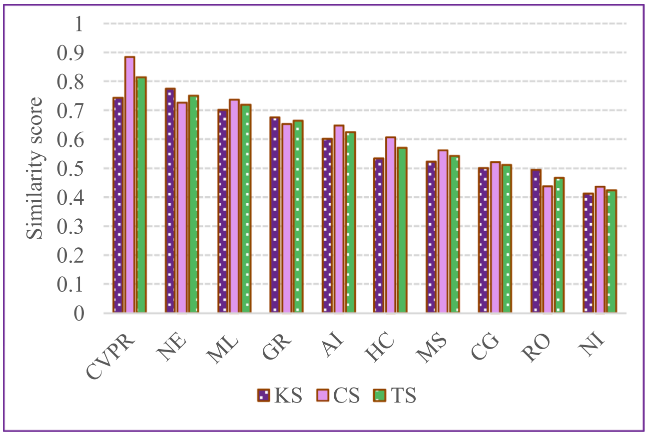


Fig. 8. Effect of content similarity on topic score (KS: Keyword Similarity CS: Content Similarity TS: Topic Similarity).

(RO), and Networking and Internet Architecture (NI). Similarity rates of both keywords and content are given. CVPR is the most appropriate title when articles in the session are tagged manually. Looking at similarity scores, keyword similarity is higher for NE than for CVPR. When only keywords are taken into account, the session title is given as NE. However, according to the similarity scores of the selected key phrases, higher similarity is obtained for the CVPR topic. Therefore, topic similarity has been calculated by averaging the keyword and content similarities. Based on the topic similarity score, CVPR is the most similar topic for the session title. As can be seen, taking into account the similarity of content has enabled more accurate titles to be chosen for the sessions.

Table 6
Real program of ICTAI.

	Timeslot 1	Timeslot 2	Timeslot 3
Day 1	Foundation −1	Foundation −2	Foundation −3
	Planning-1	Planning-2	Planning-3
	Machine Learning-1	Machine Learning-2	CVPR-1
	Neural Nets-1	Neural Nets-2	Machine Learning-3
Day 2	Foundation −4	Foundation −5	Natural Language-1
	Application Specific-1	Application Specific-2	Application Specific-3
	Machine Learning-4	Machine Learning-5	CVPR-4
	CVPR-2	CVPR-3	KRRC-1
Day 3	Natural Language-2	Decision Systems-2	
	Decision Systems-1	Decision Systems-4	
	Uncertainty	Decision Systems-3	
	KRRC-2	Data Mining	

4.5. Session scheduling results

In this subsection, an approach that takes parallel sessions into account is proposed to place sessions on appropriate days and timeslots. In order to apply the method in a real conference program, the number of days, the number of time slots, the number of parallel sessions and the links between the sessions are required. This information is given in Table 2. The proposed scheduling approach has been first tested using the ICTAI real conference program. Table 6 shows the real program of ICTAI. Looking at the real program, it is seen that the sessions on the same topic are parallel. For example; Decision Systems-2, Decision Systems-3 and Decision Systems-4 are scheduled at timeslot 2 on day 3 of the conference. According to this program, a participant who wants to attend the sessions on Decision Systems will only be able to attend one of the 3 parallel sessions and will miss the opportunity to listen to the studies in the other 2 sessions. Therefore, it is undesirable for the sessions on the same topic to be parallel, and eliminating this parallelism is the main purpose of the proposed method. For this purpose, the program has been prepared again with the proposed method using the sessions in the real program. The proposed conference program is shown in Table 7. Looking at this program, Decision Systems sessions are planned to take place on different days and timeslots. There is no conflict in the program prepared with the proposed method. Sessions on the same topic are at different times, so that participants could attend all sessions on the topic they wanted. In this way, the conference program is planned more efficiently thanks to the proposed scheduling method.

4.6. Proposed conference program results

In the proposed method, the conference program creation process is based on article similarities. Accurate determination of similarity scores directly affects other phases. The next phases are applied to the sessions created based on article similarities. The proposed approach for detecting similarities is based on the SBERT method. This method captures semantic and syntactic relations at a high rate in document representation. Therefore, it gives more successful results compared to other representation methods. The SBERT method has been compared with the Word2vec and Glove methods to demonstrate its success in creating sessions. In the Table 8, there is a sample session on the topic of Computer Vision and Pattern Recognition, which is created based on article similarities calculated by three methods. The session titles of the articles in these sessions in the real program are examined. All the articles in the session created based on SBERT are included in the CVPR session in the real program. While two of the articles in the session created based on Glove belong to a different session, three articles in the session created based on Word2vec belong to a different session. As can be seen, the sessions are more successful as a result of calculating more accurate similarities with SBERT.

The conference program, which was carried out in four stages with the proposed method, is given in the Table 9. The number of sessions in the proposed program is equal to that in the real program. The number of articles in each session is 7, as in the real program. However, since the number of articles is not a multiple of 7, the remaining articles are distributed to the sessions with the most similar articles. Standard titles are chosen for the proposed program. While the number of the same titles was higher in the real program, more topic-focused titles were determined in the proposed program. Although sessions with the same topic are placed in parallel in the real program, this parallelism does not exist in the proposed program. With the program created in general, an improvement has been achieved in the real program.

Table 10 shows the comparison of session similarity scores of ICTAI, IEEE SMC, ICDABI and ASONAM conferences. The IEEE SMC conference is the most comprehensive conference. It has articles in many fields in computer science. Since the topics of the articles are relatively less similar to each other, they have been grouped more successfully. Therefore, the similarity score of the sessions in the proposed program is higher than the other conferences. ASONAM, on the other hand, is narrower scope than ICTAI. The topic of the articles is more similar to each other. An article can be included in two different topics. Therefore, the session similarity scores of the real and proposed programs are closer to each other. Since the ICDABI conference accepted articles on different topics compared to ASONAM, the grouping of articles is more successful. Looking at the results for the four conferences, sessions have been created with higher similarity with the proposed program than with the real program.

Table 7
Program with created proposed scheduling method.

	Timeslot 1	Timeslot 2	Timeslot 3
Day 1	Planning-1 Foundation-1 Machine Learning-1 CVPR-1	Foundation −2 Machine Learning-2 CVPR-2 Decision Systems-1	Foundation −3 Machine Learning-3 CVPR-3 Decision Systems-2
Day 2	Foundation-4 Machine Learning-4 CVPR-4 Decision Systems-3	Foundation −5 Machine Learning-5 Decision Systems-4 Planning-2	Planning-3 Application Specific-1 Neural Nets-1 Natural Language-1
Day 3	Application Specific-2 Neural Nets-2 Natural Language-2 KRRC-1	Application Specific-3 KRRC-2 Uncertainty Data Mining	

Table 8
CVPR sessions created with three methods.

Method	Session papers
SBERT	■ User-Guided Image Inpatinting with Transformer ■ Terroristic Content Detection using a Multi-scene classification system ■ Bilateral Res-Unet for Image Colorization with Limited Data via GANs ■ Delving into the Scale Variance Problem in Object Detection ■ Object Quality Guided Feature Fusion for Person *Re*-identification ■ PSG-GAN: Progressive Person Image Generation with Self-Guided Local Focuses ■ Self-Augmentation with Dual-Cycle Constraint for Unsupervised Image-to-Image Generation
Glove	■ A Supervisory Mask Attentional Network for Person *Re*-Identification in Uniform Dress Scenes ■ DHQN: a Stable Approach to Remove Target Network from Deep Q-learning Network ■ Self-Augmentation with Dual-Cycle Constraint for Unsupervised Image-to-Image Generation ■ Bilateral Res-Unet for Image Colorization with Limited Data via GANs ■ Channel-Weighted Squeeze-and-Excitation Networks For Epileptic Seizure Detection ■ Delving into the Scale Variance Problem in Object Detection ■ PSG-GAN: Progressive Person Image Generation with Self-Guided Local Focuses
Word2vec	■ Cross-View Gait Recognition Based on Feature Fusion ■ Object Quality Guided Feature Fusion for Person *Re*-identification ■ Self-Augmentation with Dual-Cycle Constraint for Unsupervised Image-to-Image Generation ■ PSG-GAN: Progressive Person Image Generation with Self-Guided Local Focuses ■ Automatic Drone Identification Through Rhythm-based Features for the Internet of Drones ■ Fashion Landmark Detection via Deep Residual Spatial Attention Network ■ An Interpretation of Convolutional Neural Networks for Motif Finding from the View of Probability

Table 9
Program with created proposed scheduling method.

	Timeslot 1	Timeslot 2	Timeslot 3
Day 1	Machine Learning-1 Computer Vision and Pattern Recognition-1 Artificial Intelligence-1 Neural and Evolutionary Computing-1	Machine Learning-2 Computer Vision and Pattern Recognition-2 Artificial Intelligence-2 Information Retrieval-1	Machine Learning-3 Computer Vision and Pattern Recognition-3 Artificial Intelligence-3 Neural and Evolutionary Computing-2
Day 2	Machine Learning-4 Computer Vision and Pattern Recognition-4 Artificial Intelligence-4 Data Mining-1	Machine Learning-5 Information Retrieval-2 Neural and Evolutionary Computing-3 Data Mining-2	Social and Information Networks-1 Information Retrieval-3 Data Mining-3 Software Engineering-1
Day 3	Social and Information Networks-2 Software Engineering-2 Robotics Multiagent Systems	Social and Information Networks-3 Computer Science and Game Theory Multimedia Data Structures and Algorithms	

Table 10
Comparison of session similarity scores of ICTAI, IEEE SMC, ICDABI and ASONAM conferences.

	Real program	Proposed program
ICTAI	0.804	0.897
IEEE SMC	0.876	0.915
ICDABI	0.851	0.901
ASONAM	0.851	0.878

5. Discussion

Organizing conference programs is a difficult and time-consuming process. To facilitate this process, an approach based on article similarities is proposed. Article similarities affect the whole process. Therefore, the success of the system greatly depends on article similarities. The use of the SBERT method in the proposed approach for article similarities outperforms compared to Word2vec and Glove methods. In the calculation of article similarity, the reference part has a positive effect on success. While the similarity score for TKA is 90%, it is 93% for TKAR. The most successful results are obtained for TKAR with SBERT. In the proposed approach for creating sessions, the articles are grouped so that each session has an equal number of articles. While the average similarity score of the sessions in the real program is 80%, the average similarity score of the sessions in the proposed program is 90%. 10% improvement is achieved in the similarity of real program sessions. In the proposed approach for determining session titles, both keyword and content similarities between the articles and topics in the session are taken into account. In keyword extraction, more accurate and higher scoring keywords are extracted with the BERT-based KeyBERT algorithm compared to YAKE and TextRank algorithms. In addition, taking into account the content similarities enables the determination of more appropriate session titles. Finally, in the proposed approach for

Algorithm 1
Proposed clustering method.

```
Input: N, S, csize
Output: C = {C1, C2, …, Ck}
BEGIN
pids ← [ ]
for i ← 0 to N do
  pids[i] ← i
while pids.size > csize and clusters.size > 0 do
  for i ← 0 to N do
    if In(i, pids) then
      for j ← 0 to N do
        if In(j, pids) then
          S[i][j] ← similarity(pi, pj)
          S[j][i] ← S[i][j]
  clusters ← [ ]
  for k ← 1 to N do
    cl ← [ ]
    for i ← 0 to N do
      for j ← 0 to N do
        if S[i][j] ≥ maxValue then
        maxValue ← S[i][j]
        rindex ← i
        cindex ← j
  for i ← 0 to N do
    if i != rindex and i != cindex then
      t = average(S[rindex][i], S[cindex][i])
      S[cindex][i] ← t
      S[i][cindex] ← t
      S[rindex][i] ← 0
  S[i][rindex] ← 0
  for each c ∈ cl
    if In(rindex, c) then
      rf ← true
      rc ← index(c, cl)
    if In(cindex, c) then
        cf ← true
          cc ← index(c, cl)
  if rf = true and cf = false then
    cl[rc].append(cindex)
  if rf = false and cf = true then
      cl[cc].append(rindex)
  if rf = true and cf = true then
      merge(cl[rc], cl[cc])
      remove(cl[rc], cl)
      remove(cl[cc], cl)
  else
      cl.append(rindex, cindex)
  for each c ∈ cl
    if c.size = csize then
      clusters.append(c)
      C.append(c)
      remove(c, cl)
  pids ← [ ]
  for each c ∈ cl
    for each p ∈ c
      pids.append(p)
C.append(cl)
return C = {C1, C2, …, Ck}
END
```

scheduling sessions, the problem of parallel sessions conflicting has been resolved. A more efficient program is achieved when the real conference sessions are planned with the proposed scheduling approach. Sessions with the same title that conflicted in the real program do not conflict in the proposed program. Looking at the overall results, a more efficient program for the ICTAI conference is created with the proposed approach. As it can be seen from Table 9, a program is presented that includes sessions with a high similarity in same topic and with more specific topic titles, and where parallel sessions do not conflict. When we look at the programs presented for other conferences, there are programs in which session similarities change according to the scope of the conference, but improvements have been achieved.

6. Conclusion

Conferences are one of the important platforms that allow the sharing of articles with researchers and the emergence of new ideas. As the number of articles accepted at conferences increases and the scope of the conference expands, it becomes more difficult to plan presentation programs. In order to facilitate this process, the main purpose of this study is to automatically organize the programs prepared manually by the organizers. Programs that are not prepared effectively can have a negative impact on both presenters, participants and conference owners. That's why the conference program needs to be as efficient as possible. Getting both session content and session scheduling correct ensures that the conference achieves its main purpose. For this purpose, a four-step process to organizing conference programs is proposed. By finding similarities of the articles with the high-performance SBERT method, the probability that sessions include articles on the common topic has been increased. With the proposed clustering method, each session includes an equal number of articles. The topic of each session is determined based on both keyword and content similarities with the proposed method. A scheduling approach is presented to ensure that sessions with the same topic are not parallel. Thanks to these methods, sessions consist of articles with high similarity and in common topics. Presentation times are better managed by placing equal articles in sessions. By choosing the appropriate titles for the session articles, it is ensured that the participants can participate in the sessions based on their interests. Sessions with the same title are not placed in parallel, giving the participants the opportunity to listen to all the articles on the topic they want. An improvement is achieved with the proposed method compared to the real conference programs. As a result, with this study, a more efficient program is created. It is the first study that presents a solution to the stated problem.

CRediT authorship contribution statement

Esra Gündoğan: Conceptualization, Data curation, Investigation, Formal analysis, Software, Writing – original draft, Writing – review & editing. **Mehmet Kaya:** Conceptualization, Methodology, Investigation, Formal analysis, Supervision, Writing – original draft, Writing – review & editing, Funding acquisition.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by Scientific Research Projects Coordination Unit of Fırat University under Grant No: MF.20.09.

References

Afzali, M., & Kumar, S. (2019). Text document clustering: Issues and challenges. In *Proceedings of the international conference on machine learning, big data, cloud and parallel computing (COMITCon)* (pp. 263–268). IEEE.

Arik, O. A. (2022). *Genetic algorithm application for permutation flow shop scheduling problems, 35*. Gazi University Journal of Science.

Arunarani, A. R., Manjula, D., & Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems, 91*, 407–415.

Beliga, S., Meštrović, A., & Martinčić-Ipšić, S. (2015). An overview of graph-based keyword extraction methods and approaches. *Journal of Information and Organizational Sciences, 39*(1), 1–20.

Beniwal, A., Roy, G., & Durga Bhavani, S. (2020). Text document clustering using community discovery approach. In *Proceedings of the international conference on distributed computing and internet technology* (pp. 336–346). Springer.

Cao, T.H., .Ngo, V.M., .Hong, D.T., .& Quan, T.T. (.2018). Semantic document clustering on named entity features. arXiv preprint arXiv:1807.07777.

Chen, Y., Wang, J., Li, P., & Guo, P. (2019). Single document keyword extraction via quantifying higher-order structural features of word co-occurrence graph. *Computer Speech & Language, 57*, 98–107.

Cozzolino, I., & Ferraro, M. B. (2022). Document clustering. *Wiley Interdisciplinary Reviews: Computational Statistics*, e1588.

De, S. (2022). An efficient technique of resource scheduling in cloud using graph coloring algorithm. In *Proceedings of the global transitions*.

Deogratias, E. (2022). Using graph coloring for effective timetable scheduling at ordinary secondary level: Using graph coloring for effective timetable scheduling. *International Journal of Curriculum and Instruction, 14*(2), 1166–1188.

Devlin, J., Chang, M.W., .Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv: 1810.04805.

Domoto, K., Utsuro, T., Sawada, N., & Nishizaki, H. (2016). Spoken term detection using svm-based classifier trained with pre-indexed keywords. *IEICE Transactions on Information and Systems, 99*(10), 2528–2538.

Fortunato, S., Bergstrom, C. T., Börner, K., Evans, J. A., Helbing, D., Milojević, S., et al. (2018). Science of science. *Science, 359*(6379) (New York, N.Y.)eaao0185.

Gong, H., Sakakini, T., Bhat, S., & Xiong, J. (2019). Document similarity for texts of varying lengths via hidden topics. arXiv preprint arXiv:1903.10675.

Guleria, A., Sood, R., & Singh, P. (2021). Automatic keyphrase extraction using SVM. *Advances in communication and computational technology* (pp. 945–956). Singapore: Springer.

Gündoğan, E., & Kaya, M. (2019). Evaluation of session-suitability of papers in conference programs. In *Proceedings of the international artificial intelligence and data processing symposium (IDAP)* (pp. 1–5). IEEE.

Gündoğan, E., & Kaya, M. (2020). Research paper classification based on Word2vec and community discovery. In *Proceedings of the international conference on decision aid sciences and application (DASA)* (pp. 1032–1036). IEEE.

Kathiravan, A. V., & Kalaiyarasi, P. (2015). Sentence-similarity based document clustering using birch algorithm. *The International Journal of Innovative Research in Computer and Communication Engineering*, 3.

Khan, S., Shakil, K.A., .& Alam, M. (2016). Cloud-based big data management and analytics for scholarly resources: Current trends, challenges and scope for future research. arXiv preprint arXiv:1606.01808.

Kongwudhikunakorn, S., & Waiyamai, K. (2020). Combining distributed word representation and document distance for short text document clustering. *Journal of Information Processing Systems, 16*(2), 277–300.

Laskar, M. T. R., Huang, X., & Hoque, E. (2020). Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of the 12th language resources and evaluation conference* (pp. 5505–5514).
Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the international conference on machine learning* (pp. 1188–1196). PMLR.
Liu, J., Cheng, Q., Wang, Y., Yang, C., Zhou, R., Zhu, X., et al. (2022). An improved genetic algorithm-based traffic scheduling model for airport terminal areas. *Journal of Sensors*, 2022.
Liu, M., Lang, B., Gu, Z., & Zeeshan, A. (2017). Measuring similarity of academic articles with semantic profile and joint word embedding. *Tsinghua Science and Technology, 22*(6), 619–632.
Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., & Chi, Y. (2017). Deep keyphrase generation. arXiv preprint arXiv:1704.06879.
Murtagh, F., & Contreras, P. (2017). Algorithms for hierarchical clustering: An overview, II. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7*(6), e1219.
Nandal, P., Satyawali, A., Sachdeva, D., & Tomar, A. S. (2021). Graph coloring based scheduling algorithm to automatically generate college course timetable. In *Proceedings of the 11th international conference on cloud computing, data science & engineering (Confluence)* (pp. 210–214). IEEE.
Nguyen, T. D., & Kan, M. Y. (2007). Keyphrase extraction in scientific publications. In *Proceedings of the international conference on asian digital libraries* (pp. 317–326). Springer.
Nyandag, B. E., Li, R., & Demberel, O. (2017). Keyword extraction based on statistical information for Cyrillic Mongolian script. In *Proceedings of the 29th Chinese control and decision conference (CCDC)* (pp. 2250–2255). IEEE.
Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).
Qaiser, S., & Ali, R. (2018). Text mining: Use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications, 181*(1), 25–29.
Ravindran, R. M., & Thanamani, A. S. (2015). K-means document clustering using vector space model. *Bonfring International Journal of Data Mining, 5*(2), 10–14.
Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.
Sarkar, S., Roy, A., & Purkayastha, B. S. (2014). A comparative analysis of particle swarm optimization and K-means algorithm for text clustering using Nepali Wordnet. *International Journal on Natural Language Computing (IJNLC), 3*(3).
Squires, M., Tao, X., Elangovan, S., Gururajan, R., Zhou, X., & Acharya, U. R. (2022). A novel genetic algorithm based system for the scheduling of medical treatments. *Expert Systems with Applications*, Article 116464.
Steinert, L., & Hoppe, H. U. (2016). A comparative analysis of network-based similarity measures for scientific paper recommendations. In *Proceedings of the 3rd European network intelligence conference (ENIC)* (pp. 17–24). IEEE.
Wang, Y., & Zhang, J. (2017). Keyword extraction from online product reviews based on bi-directional LSTM recurrent neural network. In *Proceedings of the IEEE international conference on industrial engineering and engineering management (IEEM)* (pp. 2241–2245). IEEE.
Yoo, Y., Heo, T. S., Park, Y., & Kim, K. (2021). A novel hybrid methodology of measuring sentence similarity. *Symmetry, 13*(8), 1442.
Younge, K.A., .& Kuhn, J.M. (.2016). Patent-to-patent similarity: A vector space model. Available at SSRN 2709238.
Zhang, H., Zhou, X. D., & Liu, C. L. (2017). Keyword spotting in handwritten chinese documents using semi-markov conditional random fields. *Engineering Applications of Artificial Intelligence, 58*, 49–61.
Zhang, M., Li, X., Yue, S., & Yang, L. (2020). An empirical study of TextRank for keyword extraction. *IEEE Access : Practical Innovations, Open Solutions, 8*, 178849–178858.
Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.