FPSRS: a fusion approach for paper submission recommendation system

Son T. Huynh¹,²,³ · Nhi Dang¹,²,³ · Dac H. Nguyen¹ · Phong T. Huynh¹ · Binh T. Nguyen¹,²,³

Accepted: 24 August 2022 / Published online: 17 October 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Recommender systems have been increasingly popular in entertainment and consumption, and they are evident in academics, especially for applications that suggest submitting scientific articles to scientists. However, due to the various acceptance rates, impact factors, and rankings in different publishers, searching for a proper venue or journal to submit a scientific work usually takes a lot of time and effort. In this paper, we aim to present two new approaches extended from our paper (Huynh et al. 2021) presented at IAE/AIE 2021. In the first approach, we employ RNN structures besides using Conv1D. In the second approach, we introduce a new method, using DistilBert for two cases of uppercase and lowercase words. It can help vectorize features (such as Title, Abstract, and Keywords) and then use Conv1d to perform feature extraction. Furthermore, we propose a new calculation method for similarity score for Aim & Scope with other features (we name this approach is DistilBertAims). It can help keep the weights of similarity score calculation continuously updated and then continue to fit more data. The experimental results show that the second approach could obtain a better performance, which are 62.46%, 90.32%, 94.89%, 97.96% while the best performance in previous studies barely gained 50.02%, 78.89%, 86.27%, 93.23% in terms of Top K Accuracy (K = 1, 3, 5, 10). Interestingly, our best approach in this paper is higher than 12.44% the best of the previous study (Huynh et al. 2021) in terms of the Top 1 accuracy, which was presented in the conference IEA/AIE 2021.

Keywords Recommender system · RNN · DistilBert

1 Introduction
Nowadays, large-scale corporations are increasingly focusing on applying recommendation systems in human life. By diving into historical user data, these big companies can use recommendation algorithms to give the customers more excellent suggestions or understanding. It can enhance users’ experience and help raise the profit of these companies. Hence, Google, Facebook, Amazon, eBay, Spotify, and Netflix have invested human resources and money to improve and alleviate recommendation algorithms for various company products.

Interestingly, various companies are constructing recommendation systems that people can utilize in education. However, especially in the scientific academic, when junior scientists want to submit their papers, they always wonder to which publisher they should submit their research works. Consequently, these scientific papers are mistakenly submitted to journals or conferences, leading to rejection and wasting the time of both authors and reviewers. For this reason, we are motivated to study this problem to help the special masters or postdoctoral scientists who have just stepped on the path of scientific research. They can easily submit their scientific work quickly and accurately with such applications.

Our research goal is to construct a system that can give researchers the most relevant journals or conferences when receiving the basic information from manuscripts. Those are Title (T), Abstract (A), Keywords (K), and other features. In 2021, Son et al. created a state-of-the-art of this problem by using Aims and Scope as an extra feature besides Title, Abstract, and Keywords and proposing a
Convolution 1D architecture for extracting features [12]. In addition, they presented a recommendation system to recommend for researchers which journals of conferences they should submit to increase the chance of acceptance from publishers. Furthermore, they defined a method using FastText as an embedding matrix and a Convolutional one-dimension (Con1D) structure for extracting features from three inputs. Additionally, they introduced a new feature: Aims and Scope for the paper submission recommendation problem. Using FastText as an embedding layer for both Aims & Scope and other features, the system could calculate the similarity of these two types of information.

This paper presents that Aims and Scope are more effective than Title, Abstract, and Keywords in constructing the paper submission recommendation system. In addition, using Aims and Scope in combination with existing features can increase the model performance compared to not using Aim and Scope. The best performance in our work is 0.6246, 0.9032, 0.9489, and 0.9766 in Accuracy@K (K = 1, 3, 5, 10) when using Title, Abstract, and Keywords combined with the feature that we extract from Aims and Scope.

It is worth noting that the previous approach presented at IEA/AIE 2021 barely used FastText as an Embedding layer and Con1D as a primary architecture to extract features. After that, the authors applied the cosine similarity to calculate the matching value between the manuscript and each journal. Therefore, in experiments, we consider this approach as “the baseline method”. Furthermore, computing the cosine similarity of Aims and Scope with each article is not so good as the weights are not updated after each training epoch. To resolve these main limitations of this state-of-the-art, we have two approaches:

(a) **The first approach:** We use RNN’s family, including LSTM [10], BiLSTM, GRU [5], and BiGRU, instead of Convolution 1D architecture as a feature extractor for the information of a scientific paper. It is because we expect sequence architectures such as LSTM and GRU can be superior to Convolution 1D in understanding the context of sentences in each scientific paper. However, the experimental result shows that the performance on Accuracy@K (K = 1, 3, 5, 10) is worse than the baseline method compared to the second approach.

(b) **The second approach:** Due to the weakness of the first approach compared to the baseline method, Con1D holds a better performance than RNN. As a result, we focus on the embedding layer by replacing FastText embedding with DistilBert [23] models to increase the ability to understand the input information of scientific papers. In addition, it is vital to have a more efficient redesign of the cosine similarity calculation using Aims and Scope with other information. We apply DistilBert as an embedding layer for Aims and Scope to resolve this problem. Our model can then update the weights through training. We realize using DistilBert uncased without Aims & Scope can gain accuracy in terms of Top 1, Top 3, Top 5, and Top 10 are 0.5537, 0.8409, 0.9010, and 0.9524, correspondingly, compared to 0.4898, 0.7881, 0.8707, 0.9442 in the state-of-the-art methods. Furthermore, the DistilBert cased model can perform better in two cases, including Aims & Scope and without Aims & Scope. The experimental results illustrate that the proposed techniques can gain 0.5891 in Top 1 accuracy without Aim & Scope and 0.6246 in Top 1 accuracy when using Aim & Scope. Therefore, we represent this method as DistilBertAims.

Ultimately, the contributions of our work can be summarized as follows:

(i) We explore the impact of RNN’s architecture when replacing Con1D as a feature extractor in our problem. The result shows that Aims and Scope do not improve our system when combined with RNN’s structure.

(ii) We propose a second approach by replacing FastText with DistilBert as a new word embedding. This approach shows that the performance overcomes all previous state-of-the-art.

(iii) We provide a new method to calculate the similarity of Aims and Scope with other features such that the weights are always updated after each training epoch.

The paper can be organized as follows. First, Section 2 presents the associated related work of the main problem. Then, we provide the preliminaries for the main problem in Section 3 and show our two new approaches in Section 4. Next, we depict all experimental results and datasets in Section 5 and further discussions. The paper ends with our conclusion and future works.

### 2 Related work

Due to its essential role, the publication recommendation system is a crucial topic for researchers and publishers. IEEE, Springer, and Elsevier are familiar publishers who provide those practical systems for researchers to find suitable venues for their works. With a common approach, those systems take these typical features, including title, abstract, and/or keywords (or main topics) of submission as input and list out all top matched conferences or journals. However, those publishers’ recommendation systems only provide a list of conferences and journals already in their database. This action makes one step back on the
progress of publishing or sharing researchers’ ideas. Besides production-level systems, many research works related to the submission recommendation have recently been published. For instance, Feng et al. [8] studied this topic using a large dataset that included 880165 publications from 1130 different PubMed Central (PMC) journals. To extract essential features from each paper’s abstract, they used pre-trained word2vec embeddings and convolutional neural networks. Subsequently, they trained a recommendation engine with a fully linked Softmax model.

Pradhan and his colleagues [19] developed a novel scholarly venue recommendation system, the DISCOVER system, by merging social network analysis with contextual similarity. They offered the following elements for the recommendation problem: centrality measure computation, subject modeling based on contextual similarity, citation and co-citation analysis, and key-route identification based on primary path analysis of a bibliographic citation network. In terms of precision@k, stability, nDCG@k, average venue quality, variety, accuracy, MRR, and F-scores, their suggested method may surpass previous recommendation techniques in the Microsoft Academic Graph (MAG) dataset. They then expanded their technique [20] by combining a rank-based fusion of paper-paper peer network and venue-venue peer network models for the final recommendation algorithms.

Pradhan et al. [21] continued their investigation of a venue recommendation system by employing a bi-directional LSTM (Bi-LSTM) and a Hierarchical Attention Network (HAN) to construct the corresponding recommendation model with the following information from the submission: the abstract, the title, the list of keywords, the field of study, and the authors’ historical records. The experimental findings on the DBLP-Citation-Network V11 dataset show that the suggested strategy outperformed other prior recommendation techniques in F1-score, accuracy, NDCG, MRR, average venue quality, and stability.

In 2018, Wang and colleagues [24] showed promising performance with an accuracy of 61.37% in recommending top proper conferences and/or journals with a given manuscript. Wang used Chi-square and TF-IDF as feature engineering layers and a linear regression model for building the relevant classifier. Later, with the same data, with a simple deep learning approach [11], Son et al. outweighed the performance of Wang’s approach by using MLP (Multi-layer perceptron) as a classifier instead of using logistic regression with accuracy (Top 3) of 89.07%.

Dac et al. [16] used a new technique for this problem by investigating numerous deep learning methods such as LSTM [10], GRU [5], Conv1D, and the ensemble method in 2020. Interestingly, the experimental results [16] could outperform the previous results [11] in terms of Top 1, Top 3, Top 5, and Top 10 accuracies. However, the dataset volume that Wang used has only 14012 samples, which is not large enough for reliable or highly confident results.

In 2021, Son et al. released a new dataset for this problem [12]; this dataset has 414512 scientific papers from Springer’s publisher. Significantly, they used Aims and Scopes as an extra feature for increasing the performance of the paper submission system. As a result, they could gain the best results in Top 1, Top 3, and Top 5 accuracies, which are 0.5002, 0.7889, and 0.8627, respectively. In 2022, Dac and his co-workers [18] proposed a study using the Transformers model to resolve the paper submission problem. However, they barely use three features in a scientific paper, including Title, Abstract, and list of Keywords.

Previous studies have generally focused only on an article’s available features or used current state-of-the-art techniques such as the Transformers family of architectures. Therefore, in this study, we not only use Aims and Scope as a new feature type but also use DistilBert as a critical structure in feature extraction instead of RNN and Con1D.

3 Background
This section introduces several basic concepts related to the main problem used in our proposed techniques and experiments later.

3.1 FastText
Piotr Bojanowski and his colleagues introduced FastText in 2016 [1]. FastText is an extension of Word2Vec that can fix a significant drawback of Word2Vec as it can only use words in the dataset. In addition, FastText uses a second approach by dividing the text into small chunks called n-grams for each term. So, for instance, “avocado” would be “avo”, “voc”, “oca”, “cad”, and “ado” the vector of the word “avocado” would be the sum of all these instead of training for word units in wor2vec. Therefore, it handles very well for rare word cases.

3.2 DistilBert
DistilBert [23] is a variation of Bert [6] proposed in 2018 by Jacob Devlin and his colleagues and has become a state-of-the-art method in NLP. DistilBert is lighter and faster at inference time than Bert, reaching similar performances on many downstream tasks with knowledge distillation. In addition, because of requiring a smaller computational training budget, DistilBert can train and apply on compact devices with less strong hardware power.
DistilBert use a mechanism is called distillation \[3, 9\]. This technique constructs one model that plays a role as the student and is used to reproduce the behavior of a massive model or an ensemble of a model called the teacher. DistilBert can retain 97% predictive efficiency but only uses half the downstream task parameter. This paper uses DistilBert as a pre-train model and fine-tunes it with our data in two cases, including lower and upper words.

3.3 RNN

Recurrent Neural Network (RNN) \[13, 22\] is a neural network that processes information in sequence/time-series, with preprocessing of ordered data.

3.3.1 LSTM & BiLSTM

LSTM \[10\] is a special architecture of RNN \[13, 22\] capable of learning long-term dependencies introduced by Hochreiter & Schmidhuber in 1997. LSTM consist of three gates:

- Forget gate: This gate is responsible for deciding how many words to receive before the cell state.
- Input gate: This gate plays a role in deciding how much to take from the state’s input and the previous layer’s hidden layer.
- Update gate: This gate decide how much to take from the cell state to be the output of the hidden state.

LSTM still has a vanishing gradient phenomenon but less than RNN. Moreover, when carrying information on cell state, it is seldom necessary to forget the previous cell value. Thus, this architecture has been popular and widely used because of its advantages compared to RNN.

BiLSTM is the variation of LSTM, constructed by stacking two LSTM models: one receives data in a forward manner, while the other receives data in a backward way. BiLSTMs dramatically raise the amount of information possible to the network, enhancing the context available to the model.

3.3.2 GRU & BiGRU

Introduced by Cho et al. \[5\] in 2014, GRU aims to solve the gradient vanishing problem accompanying RNN. GRU is considered a variant of LSTM as both are designed similarly. In some cases, the results may be equally good. GRU includes two gates:

- Update gate: This gate decide how much past information to forget.
- Reset gate: This gate decides what information to throw away and what new information to add.

Due to the less complex structure compared to LSTM, the training time of GRU is faster, and the performance is not too bad, so the GRU is now widely used. Similar to BiLSTM, BiGRU is also built by stacking two GRU layers.

3.4 Convolution1D

Since the first impressive official appearance in 1998 \[14\], convolutional neural networks have emerged and developed as one of the most popular deep learning models in Computer Vision. The convolutional layer, nowadays, is an integral layer of many typical foundation state-of-the-art models’ architectures.

Convolutional neural networks are space invariant artificial neural networks (SIANN). Intuitively, the convolutional layer uses kernels or filters to slide along the input features and project them on another “meaningful” dimension created by hidden feature maps. The networks can extract or understand the original input’s semantic meaning by taking the “hidden feature maps” as input for the following layers. Moreover, because the kernels (filters) of a convolutional layer scan multiple cells at once, they can express the hidden meaning of the cells and their effects on each other. Many researchers try to apply the Convolutional layer according to its adjacent cell extraction ability and its fast inference time compared to the RNN layer (event LSTM or GRU). Hence, it is known as Conv1D (a one-dimensional version) in their models’ architectures for Natural Language Processing tasks.

3.5 Cosine similarity

There are several to calculate the similarity between two documents. In this paper, we choose cosine similarity as a primary baseline method. The basis is converting the content into a vector format by the embedding method. One can get pre-trained embedding weight from several websites like Word2vec, Fasttext, etc.

In each document, we calculate the center of the text, which means we calculate the mean vector of all embedding vectors representing a word. With two center vectors $A$ and $B$ of two papers, we calculate similarity by the formula below:

$$similarity = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 \cdot \sum_{i=1}^{n} B_i^2}},$$

where $n$ is the dimension of these two vectors.
4 Methodology

This section presents a second approach for a paper recommendation system from a scientific article with a Title, Abstract, and list of Keywords. In addition, we propose a further use of Aims & Scope as a new feature for enhancing the performance of the paper recommendation system. Our system allows helping the scientists able to gain more helpful information about which venues suit their work by recommending top \( N \) journals or conferences. This system can have potential applications and become a vital tool for the scientific community. In what follows, we show how to enhance the old approach using state-of-the-art techniques. We will illustrate three methods, including:

(a) The chosen baseline method: This approach was proposed by Son et al. [12]. By using Aims & Scope as an extra feature for the state-of-the-art model in [16], it can improve the performance of the paper submission recommendation problem.

(b) The first approach: Instead of using Convolution 1D in [12], we can use variations of RNN, such as LSTM, BiLSTM, GRU, and BiGRU, to get a better insight into whether using Aim and Scope can increase the performance of the system. We run different experiments for two cases, including the model using seven combinations from T, A, and K without Aims and Scope and within Aims and Scope as an extra feature.

(c) The second approach: The new research about paper submission for recommendation system [18] has shown that transformer architecture can become effective in strengthening this system. However, in that study, the author barely uses original transformers architecture instead of small transformers architecture. Consequently, the time for training as well as inference is more time-consuming. To avoid this problem, we choose the DistilBert model as an effective solution to resolve this problem. We examine two cases within and without using Aim and Scope to demonstrate that this is an essential feature in improving the performance of our recommendation system. It is worth noting that the weakness of the baseline method and the first approach is that during the training process, the weights to calculate the similarity of features such as T, A, and K with Aims and Scope are not updated after each epoch. For this reason, it is better to have proper modifications to improve the performance quality of selected models for the main problem.

In what follows, we will describe in detail how to process the data for each approach.

4.1 Baseline

The selected baseline method in experiments is presented by Son et al. [12] at IEA/AIE 2021. The authors compared different approaches [16] and measured the impact of using the information of Aims and Scope for the main problem. Interestingly, the experimental results show that using Aims and Scope can help to enhance the overall performance. This approach becomes the state-of-the-art method for the paper submission problem when using Aims and Scope as the improved feature.

4.1.1 Data preprocessing

For constructing the baseline approach, we illustrate the data processing step by step on the input data to utilize embedding methods to extract valuable features efficiently:

1. We lowercase all text to convert all words returning to the same format.
2. We eliminate not-be-alphabet words containing trivial semantics to the problem (for instance, a word “pre-treatment” or an email “author@gmail.com”).
3. We remove single letters that likely do not have pretrained weights in well-known pretrained word vectors like FastText Common Crawl.
4. We remove words within stopwords downloaded from the Natural Language Toolkit (NLTK) and additional stopwords we define.
5. We remove unnecessary space from the beginning to the end of the text after doing the four steps above.

4.1.2 Modeling

We use FastText as an embedding for seven types of input including Title (T), Abstract (A), Keywords (K), Title + Abstract (T+A), Title + Keywords (T+K), Abstract + Keywords (A+K), Title + Abstract + Keywords (T+A+K). After that, we use Convolution 1D as a feature extractor (this feature we name extracted feature). Then, we connect it using three blocks, each consisting of one Fully Connected layer and a Dropout layer. Next, we utilize Softmax as a classifier for predicted relevant journals or conferences from a scientific paper.

Interestingly, we use Aims & Scope as an extra feature to increase the performance by using FastText for embedding the Aims & Scope. We then use the cosine similarity to calculate the similarity between Aims & Scope belonging

---

1. [https://fasttext.cc/docs/en/english-vectors.html](https://fasttext.cc/docs/en/english-vectors.html)
2. [https://gist.github.com/sebleier/554280](https://gist.github.com/sebleier/554280)
The information of a scientific paper, including Title (T), Abstract (A), and Keywords (K), is combined into seven types of features (T, A, K, T+A, T+K), A+K, T+A+K). We name each integrated feature as ContentP. ContentP will join with the Aims and Scope of each journal to calculate cosine similarity. To understand how we design the cosine similarity module, one can see Fig. 2.

4.2 The first approach

Improving the performance that Dac and his colleagues proposed in [16], Son et al. [12] contributed to this method by adding Aims and Scope to increase the performance compared to the preceding works. As a result, the technique has become the state-of-the-art method for the paper submission problem using Aims and Scope as the enhanced feature. Nonetheless, we realize that this approach barely used Convolution1D as a feature extractor. As a result, in this paper, we aim to replace Convolution1D with further advanced models for sequential data such as LSTM [10], BiLSTM, GRU [5], and BiGRU.

4.2.1 Data preprocessing

As a result of inheritance of past studies, we reuse the preprocessing techniques that Son et al. [12] employed in their paper. We add further preprocessing steps to enhance the performance of the proposed methods as follows:

1. First, we remove numeric characters and all characters which are not in the alphabet (even in case the number written in words like fifteen, six, seven, sixteen, etc. could be listed in our defined stopwords).
2. Second, we separate contacted words in case the lowercase letters stand before the uppercase letter (for instance, ArtificialIntelligence → Artificial Intelligence).
3. Next, we remove all crawling errors found in the relevant texts.
4. Then, we lowercase every word to decrease complexity.
5. Finally, we remove all English stopwords (downloaded using NLTK library in Python), our defined stopwords, and all redundant backspaces. One can see more examples in Table 1.
Table 1  Two examples for our preprocessing steps

| Before                                                                 | After                                      |
|------------------------------------------------------------------------|--------------------------------------------|
| Fifteen lambs (av. BW, 22.5±8.0 kg) randomly allotted into 3 treatments | lambs av bw kg randomly allotted treatments |
| Einstein±8.0Podolskiy±93 simultaneous-approximation-term               | einstein podolskiy rosen simultaneous approximation term |
| In this work, a theoretical approach was developed for modelling olefins diffusion in two typical zeolites, HZSM-5 and HSAPo-34. Activation barrier between large cavities and channels was determined using Lennard±8.0Jones (LJ) potentials | work theoretical approach developed modelling olefins diffusion typical zeolites hzsm hsapo activation barrier large cavities channels determined using lennard jones lj potentials |

4.2.2 Modeling

This model takes input data combined with Title, Abstract, and Keywords. After that, we utilize FastText to vectorize a text into an embedding matrix having the dimension as \( M \times 300 \). Here, \( M \) is the number of words in a text, and if the actual word count of a text is not enough \( M \), we pad the sentence after so that the length is equal to \( M \). We then feed this embedding matrix through each of four models, including LSTM, BiLSTM, GRU, and BiGRU, with 100 units to efficiently extract information from low to high levels in the experiments. Furthermore, we use two blocks: a fully-connected layer with ReLu activation and a Dropout layer with a dropout rate of 0.2. Finally, we name this flow as main_flow and use Softmax at the end of the model for the final classification. One can see more detail of this model on the right-hand side of Fig. 3.

Next, we add a module that can extract the matching score of a given paper from every considered journal or conference. As described in the left-hand side of Fig. 1, each input data contains the title, the abstract, and the list of keywords from the chosen paper. One can use them to calculate the matching score between the selected article and every journal or conference stored in the database. After computing the feature vector with the length of the number of considered journals and conferences, one can pass that vector sequentially through a fully-connected layer of 1500 units, a fully-connected layer of 1000 units, and the last fully-connected layer of 500 units. We also choose a RELU activation function and a dropout layer of the rate 0.4 behind each fully-connected layer in this flow. We then concatenate this flow with main_flow as shown in the left-hand side of Fig. 1. We pass this concatenated flow through a fully-connected layer of 500 units and a dropout layer of the rate of 0.3. Finally, we use Softmax to rank the top journals or conferences relevant to a scientific paper.

As depicted in Fig. 3, we do experiments for this approach by measuring two different models for the main problem. In the first model, we combine three flows using the Aims & Scope extraction model as shown in Fig. 1. We only use the main flow for the second model without applying the Aims & Scope extraction model for better comparison.

4.3 Our second approach

As mentioned above, Dac et al. [18] demonstrated that the transformers architecture could efficiently improve the performance of the paper submission recommendation system. Consequently, we use DistilBert as a pre-trained model for embedding the inputs. This architecture ensures that it belongs to the transformers family and is compact compared to traditional transformers models. In addition to using DistilBert as a state-of-the-art, we also investigate whether using Aims and Scope has practical effects on the system.

We utilize DistilBert [23] as an embedding layer for vectorizing input instead of using FastText in previous works [12, 16]. Moreover, we use Convolution1D with different kernel sizes to extract various features to make the system superior to various kernel sizes and initialization weights. We apply DistilBert to calculate embedding features from Aim & Scope and compute the similarity score between this feature and other features. We name this model applying DistilBert for Title, Abstract, Keywords, and Aim & Scope as DistilBertAims.

4.3.1 Data preprocessing

In the semantic understanding of DistilBert, we do not reuse the data processing step mentioned above in the first approach. In our study, the manuscript’s abstract regularly includes mathematical equations or Latex scripts, which cause various problems in feature selection. Consequently, we eliminate all phrases likely to be Latex codes or mathematical equations from a given abstract.
After cleaning the data, we split each word in the input into subwords to alleviate the burden of extended English vocabulary. Then, we add two tokens [CLS] in the head and [SEP] at the end of each list of words. Finally, each type of feature combination has a different number of token sequence lengths, as shown in Table 2.

| Features            | Max sequence length |
|---------------------|---------------------|
| Title (T)           | 128                 |
| Abstract (A)        | 512                 |
| Keywords (K)        | 128                 |
| Abstract+Keywords (A+K) | 512               |
| Title+Keywords (T+K) | 256                 |
| Title+Abstract (T+A)  | 512                 |
| Title+Abstract+Keywords (T+A+K) | 512          |

4.3.2 Modeling

In this method, we design a model to get two inputs; one is a combination of three features, including Title, Abstract, and Keywords ($input_1$). The other is Aims & Scope ($input_2$). Therefore, we have two phases in this model. One is the extraction step for $input_1$, and the other is the step that calculates the similarity of $input_1$ and $input_2$. Both phases are described in the following details:

First, the DistilBert model gets $input_1$, including Title, Abstract, and Keywords, and outputs all hidden states of the model as an embedding matrix that presents the information of $input_1$. In addition, we set the dimension of the word vector as 768. Next, this embedding matrix passes through 3 Convolution1D layers in parallel with kernel sizes are $2 \times 2$, $3 \times 3$, and $4$ respectively. We choose the number of filters in each Convolution1D layer as 200. After that, we reduce the dimension of each output after using the
convolution operator by utilizing the GlobalMaxPooling layer; we name them $\text{maxpool}_1$, $\text{maxpool}_2$, and $\text{maxpool}_3$, respectively.

We denote $\text{input}_2$ as Aims & Scope; this input includes 351 documents, equal to the number of journals in our database. As described above, we calculate the cosine score of two inputs which are $\text{input}_1$ and $\text{input}_2$, as follows: firstly, two inputs also share weights in the model DistilBert, and likewise, this model is used in phase one. However, what is different here is the DistilBert model barely outputs the last state instead of all states as phase one. As a result, the output of $\text{input}_1$ after passing through DistilBert is a vector having the dimension of 768. While the output data of the remaining input is the matrix with a size of 351, the number 351 is the number of venues we mentioned above. After that, we compute the cosine score of this vector.

![Diagram](image-url)

**Fig. 4** The architecture of our second approach using DistilBert. The left figure is the model which handle features from scientific paper including Title, Abstract, and Keywords. The right feature is the model which calculates the similarity between information of a paper and Aims & Scope.
Table 3  Several samples collected with the corresponding title, abstract, and keywords

| Title | Abstract | Keywords |
|-------|----------|----------|
| Prediction of mechanical and penetrability properties of cement-stabilized clay exposed to sulfate attack by use of soft computing methods | The authors describe a novel sensor for chlorogenic acid (CGA) detection/quantification in food samples. The photosensor is based on a composite of titanium dioxide... | Photosensor, TiO2, Acridine orange, Chlorogenic acid |

Amperometric Photosensor Based on Acridine Orange/ TiO

Similar to its effects on any type of cementitious composite, it is a well-known fact that sulfate attack has also a negative influence...

Cement-stabilized soil, Strength, Penetrability, BPNN, ANFIS, Soft computing

and matrix. The result is a calculated vector with the size of $1 \times 351$; we denote this vector as $\text{cosine score}$. This method utilizes ideas from Siamese Neural Network [2, 4]. By backpropagation during training, the weights of the last input of DistilBert are updated, and our system can learn which input pairs are close together.

Finally, we concatenate three vectors containing the feature of data to make only one unique feature vector and pass it through two blocks. Each block accommodates one fully-connected layer with $n$ nodes and one Dropout layer with a rate of 0.2, and the activation function is RELU. These two blocks have the number of nodes in the fully connected layer as 500 and 400, respectively. Ultimately, we use Softmax to compute the matching score between each journal and the selected scientific paper. One can see this comprehensively in Fig. 4.

In addition, the model that does not use Aims & Scope is built by removing the similarity calculation between feature combinations and Aims & Scope.

5 Experiments

We measure the performance of baseline models and our models on a computer with Intel(R) Core(TM) i9 9900K with eight cores, 16 threads running at 3.6 GHz with 64 GB of RAM, and an Nvidia GeForce RTX2080Ti GPU.

We use Numpy and Pandas as packages for processing and reading data in our experiments. In addition, we employ the Regex package as the primary tool to clean and normalize data in the data processing phase.

Table 5  The performance of the baseline approach contributed by Son et al. in terms of Accuracy@K ($K = 1, 3, 5, 10$) for two cases: without/with using “Aims and Scopes” [12]

| Feature | Top1  | Top3  | Top5  | Top10 |
|---------|-------|-------|-------|-------|
| T       | 0.3542| 0.6634| 0.7561| 0.8532|
| TS      | 0.4015| 0.6991| 0.7971| 0.8951|
| K       | 0.3542| 0.7861| 0.8642| 0.9508|
| KS      | 0.4284| 0.7568| 0.8519| 0.9253|
| A       | 0.4691| 0.7661| 0.8482| 0.9253|
| AS      | 0.4770| 0.7662| 0.8482| 0.9253|
| TK      | 0.4157| 0.7135| 0.8232| 0.9084|
| TKS     | 0.4475| 0.7490| 0.8302| 0.9127|
| TA      | 0.4644| 0.7613| 0.8448| 0.9233|
| TAS     | 0.4828| 0.7754| 0.8536| 0.9276|
| AK      | 0.4791| 0.7730| 0.8530| 0.9273|
| AKS     | 0.4951| 0.7830| 0.8602| 0.9304|
| TAK     | 0.4852| 0.7856| 0.8624| 0.9333|
| TAKS    | 0.5002| 0.7889| 0.8627| 0.9323|

The purpose we use bold emphasis is to make it clear that using the Aims and Scope feature increases performance at each feature combination.

Table 4  Several examples collected with the corresponding Aims and Scopes in the relevant journals

| Aims and Scope |
|----------------|
| Iranian Journal of Science and Technology, Transaction A, Science (ISTT) is devoted to significant original research articles of moderate length (not more than 20 pages in the ISTT format), in a broad spectrum of Biology, Chemistry, Geology, Mathematics, and Physics... Microfluidics and Nanofluidics is an international peer-reviewed journal that aims to publish papers in all aspects of microfluidics, nanofluidics and lab-on-a-chip science and technology.... |
Furthermore, as the dataset is enormous, we need to use a multiprocessing package that can utilize the CPU’s multithreading capabilities. Thus, it makes our model run more quickly. Finally, we use Keras as a main API for the modeling procedure.

5.1 Datasets

We comprehensively describe the dataset used in the baseline approach and two new methods for our experimenting in what follows.

During this research, to compare the performance of our proposed approach with previous work, we utterly experiment on the dataset submitted first time by Son et al. [12]. This dataset consists of scientific papers collected from the publisher Springer. According to Son and colleagues, after the data collection process, this dataset has 414512 papers. To facilitate future comparison, they divided the dataset into three sets, including a training set, test set, and validation set. The splitting ratio is 60%/20%/20%. In other words, this dataset consists of 248707 samples for the training dataset, 82902 articles for the testing dataset, and 82,903 for the validation dataset.

Each scientific paper has three features: Title, Abstract, and Keywords. One can see holistically in Table 3.

### Table 6 The performance of our first approach that uses LSTM, BiLSTM, GRU, BiGRU as feature extractors

| Method | Feature | Top1  | Top3  | Top5  | Top10 |
|--------|---------|-------|-------|-------|-------|
| LSTM   | TAKS    | 0.4825| 0.7797| 0.8641| 0.9388|
|        | TAK     | 0.4837| 0.7851| 0.8690| 0.9420|
|        | TKS     | 0.4278| 0.7301| 0.8257| 0.9178|
|        | TK      | 0.4056| 0.7045| 0.8053| 0.9029|
|        | AKS     | 0.4817| 0.7771| 0.8622| 0.9373|
|        | AK      | 0.4786| 0.7780| 0.8632| 0.9392|
|        | TS      | 0.4001| 0.6945| 0.7980| 0.9011|
|        | T       | 0.3471| 0.6242| 0.7276| 0.8417|
|        | KS      | 0.4210| 0.7200| 0.8181| 0.9113|
|        | K       | 0.3757| 0.6591| 0.7632| 0.8710|
|        | TAS     | 0.4278| 0.7301| 0.8257| 0.9178|
|        | TA      | 0.4837| 0.7851| 0.8690| 0.9420|
|        | AS      | 0.4654| 0.7610| 0.8494| 0.9312|
|        | A       | 0.4615| 0.7595| 0.8497| 0.9312|
| BiLSTM | TAKS    | 0.4830| 0.7807| 0.8648| 0.9398|
|        | TAK     | 0.4782| 0.7782| 0.8645| 0.9407|
|        | TKS     | 0.4296| 0.7293| 0.8261| 0.9169|
|        | TK      | 0.4048| 0.7030| 0.8033| 0.9013|
|        | AKS     | 0.4817| 0.7774| 0.8624| 0.9389|
|        | AK      | 0.4771| 0.7784| 0.8621| 0.9388|
|        | TS      | 0.4015| 0.6982| 0.8015| 0.9021|
|        | T       | 0.3450| 0.6220| 0.7288| 0.8394|
|        | KS      | 0.4202| 0.7219| 0.8189| 0.9112|
|        | K       | 0.3748| 0.6622| 0.7661| 0.8724|
|        | TAS     | 0.4296| 0.7293| 0.8261| 0.9169|
|        | TA      | 0.4685| 0.7673| 0.8548| 0.9350|
|        | AS      | 0.4661| 0.7570| 0.8468| 0.9290|
|        | A       | 0.4606| 0.7558| 0.8460| 0.9297|

Here, we compare the performance by Accuracy@K (K = 1, 3, 5, 10) for two cases: without/with using “Aims and Scopes” (Part 1). The purpose we use bold emphasis is to make it clear that using the Aims and Scope feature increases performance at each feature combination from the publisher Springer. According to Son and colleagues, after the data collection process, this dataset has 414512 papers. To facilitate future comparison, they divided the dataset into three sets, including a training set, test set, and validation set. The splitting ratio is 60%/20%/20%. In other words, this dataset consists of 248707 samples for the training dataset, 82902 articles for the testing dataset, and 82,903 for the validation dataset.

Each scientific paper has three features: Title, Abstract, and Keywords. One can see holistically in Table 3, in latex code. Especially, Son et al. [12] crawled the corresponding Aims and Scope on the Springer website for all journals. We measure the performance among selected models on

### Table 7 The performance of our first approach using LSTM, BiLSTM, GRU, BiGRU as feature extractors

| Method | Feature | Top1  | Top3  | Top5  | Top10 |
|--------|---------|-------|-------|-------|-------|
| GRU    | TAKS    | 0.4898| 0.7881| 0.8707| 0.9442|
|        | TAK     | 0.4854| 0.7864| 0.8704| 0.9444|
|        | TKS     | 0.4321| 0.7347| 0.8312| 0.9199|
|        | TK      | 0.4087| 0.7097| 0.8106| 0.9058|
|        | AKS     | 0.4858| 0.7847| 0.8681| 0.9417|
|        | AK      | 0.4866| 0.7899| 0.8737| 0.9450|
|        | TS      | 0.4059| 0.7013| 0.8048| 0.9047|
|        | T       | 0.3473| 0.6193| 0.7258| 0.8398|
|        | KS      | 0.4259| 0.7257| 0.8223| 0.9149|
|        | K       | 0.3757| 0.6623| 0.7649| 0.8730|
|        | TAS     | 0.4321| 0.7347| 0.8312| 0.9199|
|        | TA      | 0.4854| 0.7864| 0.8704| 0.9444|
|        | AS      | 0.4747| 0.7692| 0.8565| 0.9360|
|        | A       | 0.4689| 0.7626| 0.8537| 0.9399|
| BiLSTM | TAKS    | 0.4895| 0.7882| 0.8716| 0.9437|
|        | TAK     | 0.4881| 0.7907| 0.8737| 0.9451|
|        | TKS     | 0.4321| 0.7346| 0.8308| 0.9218|
|        | TK      | 0.4071| 0.7095| 0.8093| 0.9056|
|        | AKS     | 0.4898| 0.7867| 0.8704| 0.9436|
|        | AK      | 0.4877| 0.7869| 0.8711| 0.9449|
|        | TS      | 0.4009| 0.6957| 0.7979| 0.9005|
|        | T       | 0.3485| 0.6247| 0.7309| 0.8427|
|        | KS      | 0.4204| 0.7230| 0.8216| 0.9138|
|        | K       | 0.3806| 0.6644| 0.7665| 0.8726|
|        | TAS     | 0.4321| 0.7346| 0.8308| 0.9218|
|        | TA      | 0.4881| 0.7907| 0.8737| 0.9451|
|        | AS      | 0.4718| 0.7678| 0.8550| 0.9345|
|        | A       | 0.4692| 0.7628| 0.8526| 0.9333|

Here, we compare the performance by Accuracy@K (K = 1, 3, 5, 10) for two cases: without/with using “Aims and Scopes” (Part 2). The purpose we use bold emphasis is to make it clear that using the Aims and Scope feature increases performance at each feature combination

3https://www.springer.com/gp
existing features like Title, Abstract, and Keywords and combine the Aim & Scope. One can observe the sample of Aim & Scope as shown in Table 4.

5.2 Evaluation metrics

In our recommendation system, we use \( \text{Accuracy}@K \) as a primary metric for measuring the performance of our proposed approaches. We define the \( \text{Accuracy}@K \) according to the following mathematical formulas:

\[
\text{Accuracy}@K_i = \frac{TP@K_i + FN@K_i}{TP@K_i + TN@K_i + FP@K_i + FN@K_i}
\]

Table 8 The performance of our second approach using DistilBert cased and uncased

| Method   | Feature | Top1  | Top3  | Top5  | Top10 |
|----------|---------|-------|-------|-------|-------|
| DistilBert+ | TAKS    | 0.5537| 0.8409| 0.901 | 0.952 |
| CNN1D+uncased | TAK     | 0.5503| 0.8398| 0.8959| 0.9479|
|          | TKS     | 0.4564| 0.7645| 0.8427| 0.9179|
|          | TK      | 0.4537| 0.7638| 0.8432| 0.9174|
|          | AKS     | 0.5437| 0.8396| 0.8948| 0.9456|
|          | AK      | 0.5418| 0.8438| 0.9117| 0.9620|
|          | TS      | 0.3851| 0.6806| 0.7702| 0.8627|
|          | T       | 0.3867| 0.6843| 0.7742| 0.8660|
|          | KS      | 0.4237| 0.7271| 0.8070| 0.8876|
|          | K       | 0.4232| 0.7282| 0.8091| 0.8911|
|          | TAS     | 0.5294| 0.8219| 0.8850| 0.9417|
|          | TA      | 0.5253| 0.8203| 0.8879| 0.9447|
|          | AS      | 0.5279| 0.8186| 0.8856| 0.9440|
|          | A       | 0.4458| 0.7538| 0.8361| 0.9136|
| DistilBert+ | TAKS    | 0.6246| 0.9032| 0.9489| 0.9796|
| CNN1D+cased | TAK     | 0.5891| 0.8913| 0.9427| 0.9769|
|          | TKS     | 0.5494| 0.8571| 0.9187| 0.9635|
|          | TK      | 0.5479| 0.8561| 0.9183| 0.9651|
|          | AKS     | 0.5739| 0.8633| 0.9218| 0.9663|
|          | AK      | 0.5467| 0.8535| 0.9156| 0.9656|
|          | TS      | 0.4455| 0.7448| 0.8225| 0.9001|
|          | T       | 0.4423| 0.7456| 0.8238| 0.8994|
|          | KS      | 0.4602| 0.7648| 0.8461| 0.9214|
|          | K       | 0.4561| 0.7681| 0.8478| 0.9199|
|          | TAS     | 0.5799| 0.8636| 0.9181| 0.9620|
|          | TA      | 0.5549| 0.8470| 0.9103| 0.9609|
|          | AS      | 0.576 | 0.8648| 0.9239| 0.9674|
|          | A       | 0.5020| 0.8016| 0.8707| 0.9347|

Here, we compare the performance by \( \text{Accuracy}@K (K = 1, 3, 5, 10) \) for two cases: without/with using “Aims and Scopes”. The purpose we use bold emphasis is to make it clear that using the Aims and Scope feature increases performance at each feature combination.

\( K \) is the top of \( K \) recommended results in class \( i \).
\( TP@K \) is the number of samples actual True which is predicted True allows top of \( K \) recommended results in class \( i \).
\( TN@K \) is the number of samples actual True which is predicted False allows top of \( K \) recommended results in class \( i \).
\( FP@K \) is the number of samples actual False which is predicted True allows top of \( K \) recommended results in class \( i \).
\( FN@K \) is the number of samples actual False which is predicted False allows top of \( K \) recommended results in class \( i \).

Table 9 The performance of our approach in other machine learning models: XGBoost and MLP (Multilayer perceptron)

| Method   | Feature | Top1  | Top3  | Top5  | Top10 |
|----------|---------|-------|-------|-------|-------|
| XGBosst  | TAKS    | 0.2937| 0.5961| 0.7149| 0.8437|
|          | TAK     | 0.1102| 0.2306| 0.2898| 0.3831|
|          | TAS     | 0.2937| 0.5961| 0.7149| 0.8437|
|          | TA      | 0.0626| 0.1349| 0.1837| 0.2673|
|          | AKS     | 0.2956| 0.5981| 0.7154| 0.8441|
|          | AK      | 0.1066| 0.2202| 0.2787| 0.3736|
|          | TKS     | 0.2930| 0.5987| 0.7165| 0.8456|
|          | TK      | 0.1169| 0.2404| 0.2967| 0.3832|
|          | TS      | 0.2884| 0.5884| 0.7099| 0.8406|
|          | T       | 0.0998| 0.2069| 0.2621| 0.3489|
|          | AS      | 0.2918| 0.5945| 0.7112| 0.8407|
|          | A       | 0.0778| 0.1695| 0.2237| 0.3168|
|          | KS      | 0.2964| 0.6034| 0.7208| 0.8477|
|          | K       | 0.1138| 0.2306| 0.2825| 0.3651|
| MLP      | TAKS    | 0.3815| 0.6810| 0.7867| 0.8938|
|          | TAK     | 0.0319| 0.0817| 0.1183| 0.1909|
|          | TAS     | 0.3824| 0.6783| 0.7843| 0.8936|
|          | TA      | 0.0312| 0.0796| 0.1165| 0.1885|
|          | AKS     | 0.3852| 0.6798| 0.7854| 0.8934|
|          | AK      | 0.0312| 0.0796| 0.1165| 0.1885|
|          | TKS     | 0.3906| 0.6836| 0.7907| 0.8990|
|          | TK      | 0.0243| 0.0591| 0.0871| 0.1447|
|          | TS      | 0.3920| 0.6859| 0.7923| 0.8990|
|          | T       | 0.0216| 0.0572| 0.0840| 0.1377|
|          | AS      | 0.3824| 0.6815| 0.7882| 0.8951|
|          | A       | 0.0252| 0.0597| 0.0860| 0.1404|
|          | KS      | 0.3910| 0.6868| 0.7935| 0.9001|
|          | K       | 0.0252| 0.0597| 0.0860| 0.1404|

We choose the XGBoost model with 400 estimators, and the max depth is four as its hyperparameters. In the MLP model, we construct hidden layers: the first has 600 units, the second one contains 500 units with the ReLu activation function, and the output layer has 351 units and Softmax.
The performance of different features for the first approach using BiGRU as a feature extractor. Here, we compare the performance by Accuracy@K (K = 1, 3, 5, 10) for two cases: without/with using “Aims and Scopes”

\[
Accuracy@K = \frac{\sum_{i=1}^{N} Accuracy@K_i}{N},
\]

where N is the number of Journals in the dataset.

### 5.3 Experimental results

In this section, we compare and analyze the performance of our proposed technique (DistilBertAims) which has architecture as Fig. 4 (Table 8) with the RNN-based models’ (like LSTM, BiLSTM, GRU, and BiGRU) and Son’s model—[12] which has architecture as Fig. 3. The features that we experiment with include Title (T), Abstract (A), Keywords (K), Title+Abstract (TA), Abstract+Keywords (AK), Title+Abstract+Keywords (TAK). In addition, to prove that using Aims Scope will bring better results when not in use, we also tested seven other features, including Title+ Aims/Scope (TS), Abstract+Aims/Scope (AS), Keywords+Aims/Scope (KS), Title+Abstract +Aims/Scope (TAS), Abstract+Keywords+Aims/Scope (AKS), Title+Keywords+Aims/Scope (TKS), Title+Abstract+Keywords +Aims/Scope (TAKS), and Aims & Scope (S).

Firstly, the first approach uses variations of RNN, including LSTM, BiLSTM, GRU, and BiGRU. The experimental results indicate that the performance is almost equal to the baseline method in terms of Top 1, Top 3, Top 5, and Top 10 accuracy. Next, using Aim & Scope as an additional feature also shows that the result of using this feature is always higher than without it. This result can verify that using Aims & Scope as an extra feature can help to yield good results in most cases.

At a glance, according to Tables 5, 6, 7, 8, the performances in Top K accuracy of running models on the entire feature combination (TAKS) are stable and the highest in most cases compared to other combinations. Both Fig. 5

![Figure 5](image1)

Fig. 5 The performance of different features for the first approach using BiGRU as a feature extractor. Here, we compare the performance by Accuracy@K (K = 1, 3, 5, 10) for two cases: without/with using “Aims and Scopes”

![Figure 6](image2)

Fig. 6 The performance of different features for the first approach using GRU as a feature extractor. Here, we compare the performance by Accuracy@K (K = 1, 3, 5, 10) for two cases: without/with using “Aims and Scopes”
Fig. 7 The performance of different features for the first approach using BiLSTM as a feature extractor. Here, we compare the performance by Accuracy@K (K = 1, 3, 5, 10) for two cases: without/with using “Aims and Scopes”

Fig. 8 The performance of different features for the first approach using LSTM as a feature extractor. Here, we compare the performance by Accuracy@K (K = 1, 3, 5, 10) for two cases: without/with using “Aims and Scopes”

Fig. 9 The performance of different features for the second approach using DistilBert uncased. Here, we compare the performance by Accuracy@K (K = 1, 3, 5, 10) for two cases: without/with using “Aims and Scopes”
Fig. 10 The performance of different features for the second approach that uses DistilBert cased. Here, we compare the performance by Accuracy@K (K = 1, 3, 5, 10) for two cases: without/with using “Aims and Scopes.”

DistilBertAims and Recurrent-based neural networks’ Top 1 accuracy when processing on TAKS combination (uncased) are higher than different feature combinations (above 48%).

By using DistilBertAims as the backbone, Conv1D (multi-size kernels) as filters, adjacent with Siamese convention for the similarity score, DistilBertAims precisely categorizes the proper conferences/journals for the corresponding contextual input and generally achieves better predictions compared to the Son et al. with the highest excess efficiency for Top 1 accuracy of over 5.35% (the uncased one). With DistilBertAims as a model, taking TAKS (cased) as input’s performance dominate any other feature combinations as well as any models in most Top K accuracy (K=1, 3, 5, 10) with 62.46%, 90.32%, 94.89% and 97.96% respectively Table 8.

Additionally, we expanded our experiments to contextual feature differences. We applied DistilBertAims on two different inputs, the cased approach (keep the upper-cases) and the uncased approach (lower-cased). This exploration gives us a remarkable outcome as follows. The overall performance of DistilBertAims on cased feature combinations outperforms all other models on all different feature combinations. Furthermore, it is typically better than our DistilBertAims on uncased from 3 to 10% in all Top K accuracy.

Finally, we also run some more experiments based on the previous state-of-the-art methods by changing the classification algorithm to XGboost and MLP (one can see the result of this approach in Table 9). The results show that our method still gives better results, especially when using Aims and Scope.

In short, the DistilBertAims model can extract the implied information of the contextual input much more than typical RNN-based models and standalone transformers by combining advantages of Transformer architecture, data filtering capabilities of CNN layer, and the similarity extracted from Siamese structure. Moreover, with Aim & Scope as an additional input, DistilBertAims shows its dominant and stable performance in solving this problem.

In summary, one can see comprehensively the results in Figs. 5, 6, 7, 8, 9, and 10.

6 Conclusion and future works

We have extended the previous method [12] by using FastText as an embedding layer and Convolution 1D as a feature extractor for the paper submission recommendation system. We have contributed two approaches; one extends from our previous article by using variations of Recurrent Neural Network instead of Convolution 1D as a feature extractor. The other utilizes DistilBert for embedding layer instead of FastText for Title, Abstract, Keyword, and Aim & Scope in our old paper [12]. Furthermore, we utilize multi Convolution 1D with different kernel sizes to capture various input semantics. The result shows that the performance of this technique outperforms all previous methods. Interestingly, we also introduce a new method for calculating the similarity of Aim & Scope.

Our approach shows that using Aims and Scope is efficient in improving the performance of the paper submission problem. In the second approach, in all combinations of features, our new proposed model beat the best model in the previous state-of-the-art technique (the best model in our earlier paper). Especially in terms of Top 1 Accuracy with feature TAK, the best model in the second approach overcame 12.44% than the baseline model.

Due to the complicated structure of Transformers, even though using a smaller variant of Bert, DistilBert, this technique...
is still too large compared to other methods like Glove or FastText. The main disadvantage of the second approach is that it is time-consuming in training and inference.

In the future, we plan to experiment on a larger dataset and the various publisher instead of Springer and use other ways to calculate similarity scores. Furthermore, to reinforce the model performance in terms of accuracy and speed, we aim to extend our experiment to using a Transformer with the Fourier Transform as an alternative to Self-Attention [15]. Finally, applying the Mixture of Experts (MoE) inside the model architecture inspired by Nan Du et al. [7, 17] may enhance the completeness and the ability of the model’s architecture. It can increase the total parameters but reduce the running parameters in an inference process.

Acknowledgements Son Huynh Thanh was funded by Vingroup JSC and supported by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Institute of Big Data, code VINIF.2021.ThS.18

References

1. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146
2. Bromley J, Guyon I, LeCun Y, Säckinger E, Shah R (1993) Signature verification using a “siamese” time delay neural network. In: Proceedings of the 9th international conference on neural information processing systems, NIPS’93. Morgan Kaufmann Publishers Inc., San Francisco, pp 737–744
3. Bucilaannedefined C, Caruana R, Niculescu-Mizil A (2006) Model compression. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, KDD ’06. Association for Computing Machinery, New York, pp 535–541. https://doi.org/10.1145/1150402.1150464
4. Chicco D (2021) Siamese neural networks: an overview. Springer US, New York, pp 73–94. https://doi.org/10.1007/978-1-0716-0826-5_3
5. Cho K, van Merriënoer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, pp 1724–1734. https://doi.org/10.3115/v1/D14-1179. https://aclanthology.org/D14-1179
6. Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T (eds) Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, vol 1. Long and Short Papers. Association for Computational Linguistics, pp 4177–4186. https://doi.org/10.18653/v1/n19-1423
7. Du N, Huang Y, Dai AM, Tong S, Lepikhin D, Xu Y, Krikun M, Zhou Y, Yu AW, Firat O, Zoph B, Fedus L, Bosma A, Zhou Z, Wang T, Wang YE, Webster K, Pellatt M, Robinson K, Meier-Hellstern K, Duke T, Dixon L, Zhang K, Le QV, Wu Y, Chen Z, Cui C (2021) Glam: efficient scaling of language models with mixture-of-experts. arXiv:2112.06905
8. Feng X, Zhang H, Ren Y, Shang P, Zhu Y, Liang Y, Guan R, Xu D (2019) The deep learning–based recommender system “pubmender” for choosing a biomedical publication venue: development and validation study. J Med Internet Res 21(5):e12957. https://doi.org/10.2196/12957. http://www.jmir.org/2019/5/e12957/
9. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. In: NIPS Deep learning and representation learning workshop. http://arxiv.org/abs/1503.02531
10. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735
11. Huynh ST, Huynh PT, Nguyen DH, Cuong DV, Nguyen BT (2020) S2rscs: an efficient scientific submission recommendation system for computer science. In: Fujita H, Fournier-Viger P, Ali M, Sasaki J (eds) Trends in artificial intelligence theory and applications. Artificial intelligence practices. Springer International Publishing, Cham, pp 186–198
12. Huynh ST, Dang N, Huynh PT, Nguyen DH, Nguyen BT (2021) A fusion approach for paper submission recommendation system. In: International conference on industrial, engineering and other applications of applied intelligent systems. Springer, pp 72–83
13. Jordan MI (1997) Chapter 25 serial order: a parallel distributed processing approach. In: Donahoe JW, Packard Dorsel V (eds) Neural-network models of cognition, advances in psychology, vol 121. North-Holland, pp 471–495. https://doi.org/10.1016/S0166-4115(97)80111-2. https://www.sciencedirect.com/science/article/pii/S0166411597801112
14. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324. https://doi.org/10.1109/5.726791
15. Lee-Thorp J, Ainslie J, Eckstein I, Ontanon S (2021) Fnet: mixing tokens with fourier transforms. arXiv:2105.03824
16. Nguyen D, Huynh S, Huynh P, Dinh CV, Nguyen BT (2021) S2cft: a new approach for paper submission recommendation. In: SOFSEM 2021: theory and practice of computer science. Springer International Publishing, Cham, pp 563–573
17. Nguyen CV, Le KH, Tran AM, Pham QH, Nguyen BT (2022) Learning for amalgamation: a multi-source transfer learning framework for sentiment classification. Inf Sci 590:1–14. https://doi.org/10.1016/j.ins.2021.12.059. https://www.sciencedirect.com/science/article/pii/S0020025521012809
18. Nguyen DH, Huynh ST, Dinh CV, Huynh PT, Nguyen BT (2022) Psrnte: paper submission recommendation using mixtures of transformer. Expert Syst Appl 117096. https://doi.org/10.1016/j.eswa.2022.117096. https://www.sciencedirect.com/science/article/pii/S0957417422005024
19. Pradhan T, Pal S (2019) A hybrid personalized scholarly venue recommender system integrating social network analysis and contextual similarity. Future Gener Comput Syst. https://doi.org/10.1016/j.future.2019.11.017
20. Pradhan T, Pal S (2020) Cnaver: a content and network-based academic venue recommender system. Knowl-Based Syst 189:105092. https://doi.org/10.1016/j.knosys.2019.105092. http://www.sciencedirect.com/science/article/pii/S0957057019304691
21. Pradhan T, Gupta A, Pal S (2020) Hasvrec: a modularized hierarchical attention-based scholarly venue recommender system. Knowl-Based Syst 204:106181. https://doi.org/10.1016/j.knosys.2020.106181. http://www.sciencedirect.com/science/article/pii/S0957057020304135
22. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. MIT Press, Cambridge, pp 318–362
23. Sanh V, Debut L, Chaumond J, Wolf T (2019) Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv:1910.01108
24. Wang D, Liang Y, Xu D, Feng X, Guan R (2018) A content-based recommender system for computer science publications. Knowl-Based Syst 157:1–9. https://doi.org/10.1016/j.knosys.2018.05.001. http://www.sciencedirect.com/science/article/pii/S0950705118302107

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Son Thanh Huynh is currently a Master’s student at the Faculty of Mathematics and Computer Science, VNU HCM - University of Science. He graduated with a bachelor of science in 2020 at VNU HCM - University of Science. His major research is related to applied machine learning and recommendation systems. He is working as Senior Research Engineer at the AISIA Research Lab in Ho Chi Minh City, Vietnam.

HUU-DAC NGUYEN is Data Scientist & Quantitative Researcher at Athena and is a Research Assistant at AISIA Lab. He got his Master of Engineering Degree in 2021 at John von Neumann Institute. He was a Data Science Sub-team Lead in ADAS (Advanced Driver-Assistance Systems) at Vantix Inc., Vingroup.

Dang Nhi is currently working at Power Engineering Consulting Joint Stock Company 2 (PECC2) company as an AI engineer for solving NLP and Computer Vision problems. He graduated with a bachelor of science in 2021 at VNU HCM - University of Science.

Phong T. Huynh obtained the bachelor’s degree in Mathematics from University of Science (VNUHCM) in 2020. She is currently working as a Data engineer at DataFirst, Hung Thinh Corporation.

Son Thanh Huynh

Dr. Binh Thanh Nguyen is Head of Department of Computer Science at Faculty of Mathematics and Computer Science, VNU HCM - University of Science. He has had over ten years of experience in AI and data science. He defended his Ph.D. thesis with the highest honors at Ecole Polytechnique (Paris, France) in 2012. Up to now, he has had over 60 publications and four patents filed in USA and Canada. He also had substantial experience in building Research and Development teams for helping the company or the startup deliver AI products.

Dang Nhi

Phong T. Huynh

Dr. Binh Thanh Nguyen