Conference Management System Utilizing an LLM-Based Recommendation System for the Reviewer Assignment Problem

Vaios Stergiopoulos[1][a], Michael Vassilakopoulos[1][b], Eleni Tousidou[1][c], Spyridon Kavvathas[1] and Antonio Corral[2][d]

[1]*Data Structuring & Eng. Lab., Dept. of Electrical & Computer Engineering, University of Thessaly, Volos, Greece*

[2]*Dept. of Informatics, University of Almeria, Spain*

Keywords: Conference Management System, Large Language Models, Recommendation System, Paper-Reviewer Matching, Reviewer Assignment Problem.

Abstract: One of the most important tasks of a conference organizer is to assign reviewers to papers. The peer review process of the submitted papers is a crucial step in determining the conference agenda, quality, and success. However, this is not an easy task; large conferences often assign hundreds of papers to hundreds of reviewers, making it impossible for a single person to complete the task due to hard time constraints. We propose a Conference Management System that embodies a Large Language Model (LLM) in its core. The LLM is utilized as a Recommendation System which applies Content-based Filtering and automates the task of reviewers-to-papers assignment for a conference. The LLM we select to use is the Bidirectional Encoder Representations from Transformers (BERT), in two specific variants, BERT-tiny and BERT-large.

1 INTRODUCTION

Today, the number of papers submitted to academic conferences is continuously increasing, resulting in a phenomenon that is extremely intense in several prominent ones. For example, the number of manuscripts received by NeurIPS and AAAI conferences in 2020 was more than 5 times that in 2014 (Zhao and Zhang, 2022).

The assignment of the submitted papers (proposals) to the most scientifically qualified reviewers, known as the Reviewer Assignment Problem (RAP), remains a very crucial responsibility of conference organizers to this day, as it is an essential component of academic integrity and excellence. The quality and reputation of a conference depend greatly on the receipt of high-quality reviews. The most prestigious conferences usually have to assign thousands of papers to thousands of reviewers, and, with the added time constraint, this becomes a challenging operation to be handled by a single person or a small group of people. Therefore, systems managing the full conference procedures and assigning reviewers to articles automatically are being increasingly utilized to manage the above situation. In addition, such systems receive a lot of attention from researchers around the world (Zhao and Zhang, 2022) (Aksoy et al., 2023) (Ribeiro et al., 2023) to manage the growth in the number of articles and solve the reviewer assignment problem.

On the other hand, recent research has put a lot of focus on Large Language Models (LLMs), or Pretrained Language Models (PLMs); they are large-scale, pretrained language models based on neural networks that have been trained with a vast amount of generic textual datasets. Furthermore, recent advances on transformer-based LLMs (Vaswani et al., 2017) specifically have provided state-of-the-art performance to a wide variety of tasks and applications, ranging from chatbots to summarizing text and translating. Here, we accomplish to utilize the power of LLMs in Recommendation Systems. More specifically, we embody a LLM in a recommendation engine that deals with the RAP problem inside a web-based Conference Management System.

In this paper, we present our system in the above-mentioned framework:

- A web-based application which basically is a full

[a] https://orcid.org/0000-0003-3458-1491
[b] https://orcid.org/0000-0003-2256-5523
[c] https://orcid.org/0000-0002-1581-0642
[d] https://orcid.org/0000-0002-0069-4642

Stergiopoulos, V., Vassilakopoulos, M., Tousidou, E., Kavvathas, S. and Corral, A.
Conference Management System Utilizing an LLM-Based Recommendation System for the Reviewer Assignment Problem.
DOI: 10.5220/0013482600003929

In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 1*, pages 1004-1011
ISBN: 978-989-758-749-8; ISSN: 2184-4992

Conference Management System that can automate and complete all the sub-tasks needed by the conference organizers.

- A solution for the classic Reviewer Assignment Problem (RAP).
- An application of LLMs in Recommendation Systems (RS) as a solution for RAP.

The remainder of this paper is organized in the following order. Firstly, all related work is presented in Section 2. Secondly, the Recommendation System is explained in Section 3. Next, our web Conference Management System is demonstrated in Section 4. Lastly, in Section 5 we conclude this work and discuss related future work directions.

2 RELATED WORK

In this section, we first review related work on LLMs and Recommendation Systems and we next review work on Conference Management Systems and the reviewer assignment problem.

2.1 Large Language Models and Recommendation Systems

Vaswani et al. (Vaswani et al., 2017) propose a specific kind of deep neural network called Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Their work was initially suggested for translation tasks, but later proved to be a huge advancement for LLMs in general. Their suggested network architecture comprises encoder and decoder stacks (each one of them consisting of 6 identical layers) and is the base for all state-of-the-art LLMs today.

The BERT LLM, proposed by Devlin et al. (Devlin et al., 2019), which stands for Bidirectional Encoder Representations from Transformers, is designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers (process all tokens before and after at each tokenized word). The BERT model architecture is a multilayer bidirectional transformer encoder which computes vector-space representations of natural language that are suitable for use in deep learning models. BERT models are usually pre-trained on a large corpus of text, then fine-tuned for specific tasks. It has been successful in a variety of tasks in NLP (natural language processing). Also, the pre-trained BERT representations can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

Moreover, Sun et al. (Sun et al., 2019) introduce a new sequential recommendation model, which adopts BERT to a new task, sequential recommendation. Thus, the BERT LLM is embodied in a Recommendation System. Their system, called BERT4Rec, employs the deep bidirectional self-attention to model user behavior sequences and predicts the random masked items in the sequence by jointly conditioning on their left and right context. They train a bidirectional representation model to make recommendations by allowing each item in user historical behaviors to fuse information from both left and right sides.

Wu et al. (Wu et al., 2021) present their efforts to strengthen a news Recommendation System by utilizing pre-trained language models (PLMs). Results from offline experiments on datasets for monolingual and multilingual news recommendation demonstrate that using PLMs for news modeling can significantly enhance news recommendation performance. After being implemented on the Microsoft News platform, the PLM-powered news recommendation models saw notable increases in clicks and pageviews in both English and international markets. So, in their framework, they instantiate a news encoder PLM to capture the deep contexts in news texts. They denote an input news text with a number of tokens. Next, the PLM converts each token into its embedding, and then learns the hidden representations of words through several Transformer layers; this way a hidden token representation sequence is produced. Moreover, they use an attention network to summarize the hidden token representations into a unified news embedding which is further used for user modeling and candidate matching.

2.2 Conference Management Systems and the Reviewer Assignment Problem

The work of Charlin & Zemel (Charlin and Zemel, 2013), the "Toronto Paper Matching System" (TPMS), is one of the first high-performance attempts to create a reviewers assignment system that would automate this task for a prominent conference. They use Latent Dirichlet Allocation (LDA), an unsupervised probabilistic method used to model documents, and a matching score that predicts a reviewer's score as the dot product between a reviewer's profile representation (archive data of her/his papers) and each submission.

Furthermore, Maleszka et al. (Maleszka et al., 2020) present the idea of an overall modular system

for determining a grouping of reviewers, as well as three modules for such a system: a keyword-based module, a social graph module, and a linguistic module. They start from a single reviewer and look for a diverse group of other possible candidates that would complement the first one in order to cover multiple areas of the review.

Additionally, Stelmakh et al. (Stelmakh et al., 2023) provide publicly a gold-standard dataset that is needed to perform reproducible research on the process of assigning submissions to reviewers. Their dataset consists of 477 self-reported expertise scores provided by 58 researchers (reviewer candidates) who evaluated their expertise in reviewing papers they had previously read. Moreover, they use this data to compare several popular algorithms currently employed in conferences and present the respective performance metrics. This currently expanding dataset could be the standard dataset for algorithmic comparison in the area of RAP task.

The study of Bouanane et al. (Bouanane et al., 2024) introduces the Balanced and Fair Reviewer Assignment Problem (BFRAP), which aims to maximize the overall similarity score (efficiency) and the minimum paper score (fairness) subject to coverage, load balance, and fairness constraints. They conduct a theoretical investigation into the threshold conditions for the problem's feasibility and optimality. To facilitate this investigation, they establish a connection between BFRAP, defined over m reviewers, and the Equitable m-Coloring Problem. Building on this theoretical foundation, they propose FairColor, an algorithm designed to retrieve fair and efficient assignments.

Leyton-brown et al. (Leyton-Brown et al., 2024) in their paper introduce Large Conference Matching (LCM), a novel reviewer–paper matching approach that was recently deployed in the 35th AAAI Conference on Artificial Intelligence (AAAI 2021), and has since been adopted by other conferences. LCM consists of three primary components:

1. Gathering and managing input information to pinpoint issues with matches and produce scores for reviewers and papers;
2. Creating and resolving an optimization challenge to identify optimal reviewer-paper pairings; and
3. A two-phase evaluation process that reallocates review resources from submissions that are probable to be declined and toward documents closer to the decision limit.

Zhang et al. (Zhang et al., 2024) propose a unified model for paper-reviewer matching that jointly considers semantic, topic, and citation factors. To be specific, during training, they instruction-tune a contextualized language model shared across all factors to capture their commonalities and characteristics; during inference, they chain the three factors to enable step-by-step, coarse-to-fine search for qualified reviewers given a submission.

Latypova (Latypova, 2023) proposes a method of reviewer assignment decision support in an academic journal based on a joint use of multicriteria assessment and text mining. Calculation of an integral indicator with the use of additive folding of weighted reviewer's indicators is at the core of the method. Text mining of manuscripts and reviewer's papers is utilized to determine value of one of significant indicators. The proposed method allows to assess reviewers not only by authority and expertise, but also allows to take into account their work in the role of a reviewer, deciding how good they are in this role.

3 THE RECOMMENDATION ENGINE

In this section, we describe the functional architecture of our recommendation engine as the core part of the full Conference Management System. The recommendation engine works smoothly with the rest of the components of the system, while various procedures are executed, e.g. conference setup, reviewers input, paper submission, etc.

The recommendation engine periodically checks the system's database for new conferences. As soon as a new conference is created and the user (conference organizer) characterizes it as finalized, meaning that the submitted papers, as well as the list of the reviewers, have been inserted into it, the recommendation engine starts processing the specific conference.

To begin with, the Reviewer Module retrieves from the database the reviewers' data, i.e. full name, email and university/organization. In case a reviewer already has a profile in the database, created at a previous conference, the module just loads this profile. If a reviewer does not have a profile in our database, the module creates a new scientific profile by collecting text from all the papers she/he has coauthored, using data from the AMiner Citation Network Dataset. We have pre-processed the AMiner Citation Network Dataset in order to easily retrieve an author and her/his relevant papers. Moreover, all papers' text has been cleaned and pre-processed, to accelerate procedures and make them more efficient.

Additionally, the Reviewer Module utilizes the "edit distance measure", also known as the generalized Levenshtein distance, presented in Li et al. (Yujian and Bo, 2007), to calculate the differences be-

tween sequences and patterns. The Levenshtein distance, d(x,y), computes the minimal cost of transforming string x to string y. The transformation of a string is carried out using a sequence of the following operators: delete a character, insert a character, and substitute one character for another. We use the Levenshtein distance in order to compare the reviewers' names with the authors' names of the AMiner Citation Network Dataset and retrieve the right authors' profiles. Finally, the reviewers module provides the BERT LLM part with the reviewers' scientific profiles.

Furthermore, there is the Papers Module which processes the submitted papers and their data. It retrieves all submitted papers from every single conference in the database and executes the necessary text pre-processing and cleaning, implementing the methods described by Stergiopoulos et al. (Stergiopoulos et al., 2022). Next, the papers module outputs the preprocessed text (title, abstract) of the papers into the BERT LLM part.

In the core of our recommendation engine (Figure 1) there is the BERT LLM (Devlin et al., 2019) which computes the vector-space representations for both the reviewers (text from the researcher's profile) and the submitted papers (text from title and abstract).

To be more specific, we run two sets of experiments with two BERT variants:

1. BERT-tiny: a BERT model with 2 layers, 128 hidden unit size, and 2 attention heads (L-2, H-128, A-2), pre-trained on uncased text.
2. BERT-large: a BERT model with 24 layers, 1024 hidden unit size, and 16 attention heads (L-24, H-1024, A-16), pre-trained on uncased text.

The BERT LLM (Figure 1) usually comprises two processing layers: the preprocessor and the transformer-encoder layer.

The preprocessor layer receives as input, simple natural language text. It tokenizes, formats and packs input sentences. The result of the preprocessing layer is a batch of fixed-length input sequences of tokens for the transformer encoder. An input token-sequence starts with one start-of-sequence token, followed by the tokenized segments, each terminated by one end-of-segment token. Also, it provides the token dictionary to the next layer.

Next, the output of the preprocessor layer is inserted into the transformer-encoder layer. It doesn't just take the tokenized strings as input, but it also expects these to be packed into a particular format. The transformer-encoder layer creates and outputs the actual vector-space representations for both the reviewers and the submitted papers, so that the Reviewers

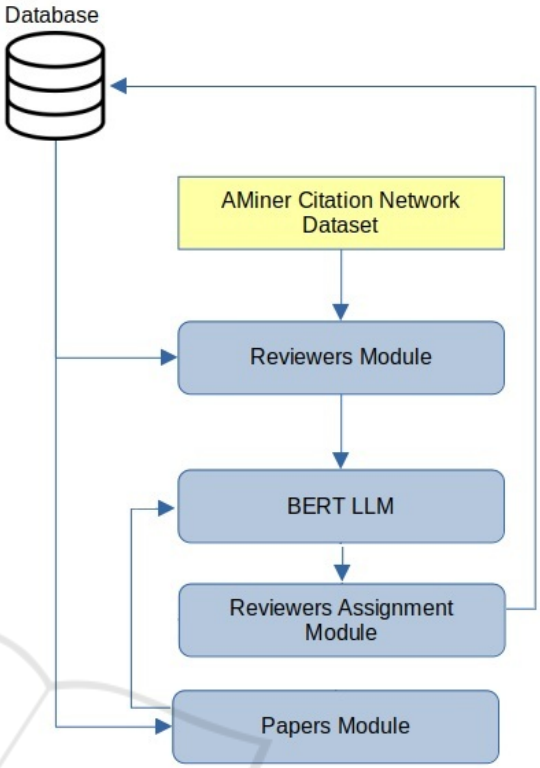

Figure 1: The functional architecture of our Recommendation Engine.

Assignment Module - RAM (Figure 1) can use them to calculate their similarity.

The RAM in Figure 2, incorporates a number of assumptions and constraints as listed below:

- Each paper must be reviewed by a specific number of reviewers; the reviewers are assigned a maximum number of papers, usually three, and the papers are assigned to three reviewers, too. This is a variable that could change depending on the user's needs.
- Reviewers have a limit on the number of papers they can review.
- The authors of submitted papers should not be reviewers at the same moment; a mechanism that increases the integrity of the peer review procedure, as the reviewers that are authors of a paper in a conference are removed from the reviewers list of the specific conference.
- Assign the most suitable reviewers, depending on their expertise, for each paper while meeting the aforementioned constraints.

Our recommendation engine actually implements Content-based Filtering (CBF), i.e. it calculates the similarity between a reviewer and a paper, trying to

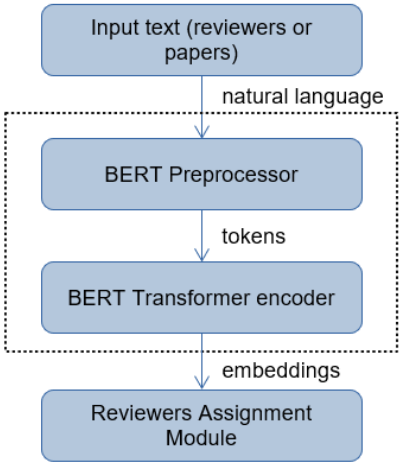

Figure 2: The BERT LLM in the core of our recommendation Engine.

predict the expertise of a reviewer for a paper. RAM uses the embeddings of both reviewers and papers and calculates the similarity between them. Actually, cosine similarity is being used as it performs really well in text similarity tasks. Cosine similarity is a measure of similarity between two vectors in an inner product space. It determines the degree to which two vectors are pointing in the same direction by calculating the cosine of the angle between them. Cosine similarity is commonly used in text analysis to measure the similarity between documents based on the frequency of words or phrases they contain. So, if this distance is small, there will be a high degree of similarity, but when the distance is large, there will be a low degree of similarity. Our aim is, via cosine similarity, to determine (predict) a reviewer's expertise on a paper.

Later on, the RAM recommends (assigns) reviewers to papers in a high-degree-of-similarity-goes-first logic. Specifically, the first paper (that is being processed) is assigned to the three reviewers who are found to have the highest similarity to it, then the second paper, and so on. Finally, it outputs the resulting reviewer-to-paper recommendations (assignments) into the database of the system, so that the user (conference organizer) can check it, make any adjustments or modifications and approve it. As soon as the reviewers recommendation is approved, the reviewers are being informed of the papers that have been assigned to them and they can start the peer review process.

4 THE WEB SYSTEM

4.1 Technologies

In this section, the design of the system is presented in order to understand in detail how the system operates. As shown in Figure 3 the system clearly consists of four different entities. Firstly the front-end, which includes the user interface, the client side routing and the integration with the external service; secondly the back-end, which includes the routing, the middleware and the data handling; thirdly the database, which provides and receives data from the back-end; last but not least, the Recommendation System, which communicates directly with the database to retrieve papers and reviewers data, or send the recommendations (assignments).

The choice of web application type was meticulously determined by prioritizing factors such as user experience, application speed, and overall fluidity. Crucial considerations were given to seamless integration between front-end, back-end, and database components. After comprehensive evaluation, it became evident that the Single Page Application (SPA) model best aligned with the objectives of our system. Leveraging the React.js framework for front-end implementation further reinforced our commitment to delivering an intuitive and responsive user interface.

The term back-end refers to the server-side of a web application responsible for handling data processing, logic, and database management. It acts as the backbone of the system, facilitating seamless communication between the front-end and the database. Here, for the web-based Recommendation System for scientific conferences, we employed a combination of powerful technologies to ensure robustness, scalability, and efficiency. The technologies utilized in the back-end include Node.js, Express.js, SQL, and MySQL.

4.2 Requirements Analysis

It is crucial to outline the various capabilities (system requirements) that users have when using this web-based Recommendation System for scientific conferences. Below we present an analytical list of the system requirements:

1. Users can create a new account using the registration page.
2. Users can log in to the application using the credentials provided during the registration process.
3. Users have the option to log out of the application at any time after registering or logging in.

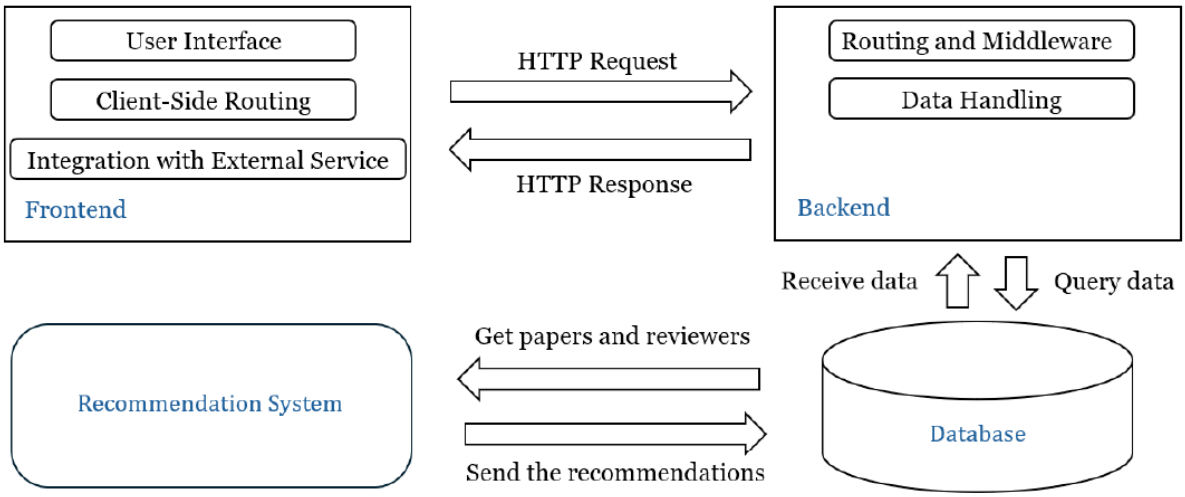

Figure 3: The architecture of the system.

4. Once logged in, users can navigate through the application's features and functionalities.
5. Users can create new scientific conferences, providing all the necessary information required for a scientific conference.
6. Users can view the conferences that have been created.
7. Users have the ability to add scientific papers in multiple ways:
   - Add papers one by one, providing only the title and abstract.
   - Upload papers individually, where the system extracts the title and abstract automatically.
   - Import papers via Excel or CSV files, allowing multiple papers to be added simultaneously.
8. Users also have the ability to add reviewers for their conferences using various methods:
   - Add reviewers one by one, providing their email and name.
   - Upload reviewers individually, with the system extracting details from the provided information.
   - Import reviewers through Excel or CSV files, facilitating the addition of multiple reviewers at once.
9. Users can automatically create the reviewers-to-papers-assignment.

4.3 Graphical User Interface

Following, we present the Graphical User Interface (GUI) of this system.

Initially, the potential user has to create an account (register) in the Home page, as shown in Figure 4, to be able to use the system.

After successful registration and login, the user can observe the options available within the application (Figure 5). So now, she/he can initiate a new conference by clicking the Create button, review and update conferences she/he has already created via the View button, and participate in other conferences by selecting the Join button (e.g. if the user is a reviewer for one or more conferences).

The management of conferences (i.e. create a new conference or update an existing one) can be accomplished via the options in the Conference page (Figure 6). There, the user can insert proposals (submitted papers) or reviewers; these procedures can be done either one by one (single paper or reviewer) or via bulk input using csv files.

Finally, the system provides the assignments of reviewers to papers, as can be seen in Figure 7. After reviewing the recommendations (assignments), the user has the option to export them, either as a CSV file or in XLSX format for further analysis or sharing.

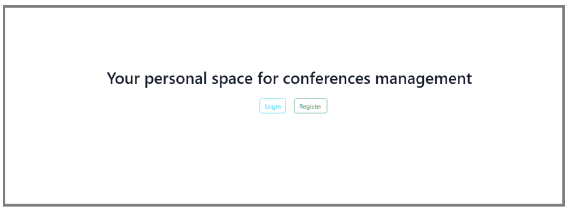

Figure 4: The Home page of the web-system.

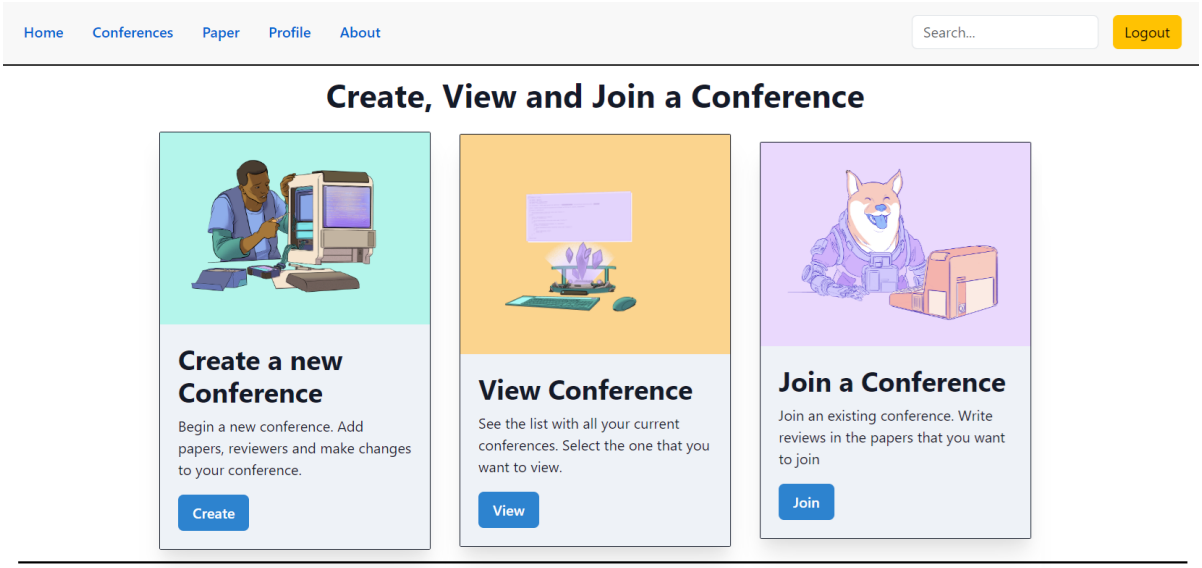

Figure 5: The basic user page of the system with all the available options.

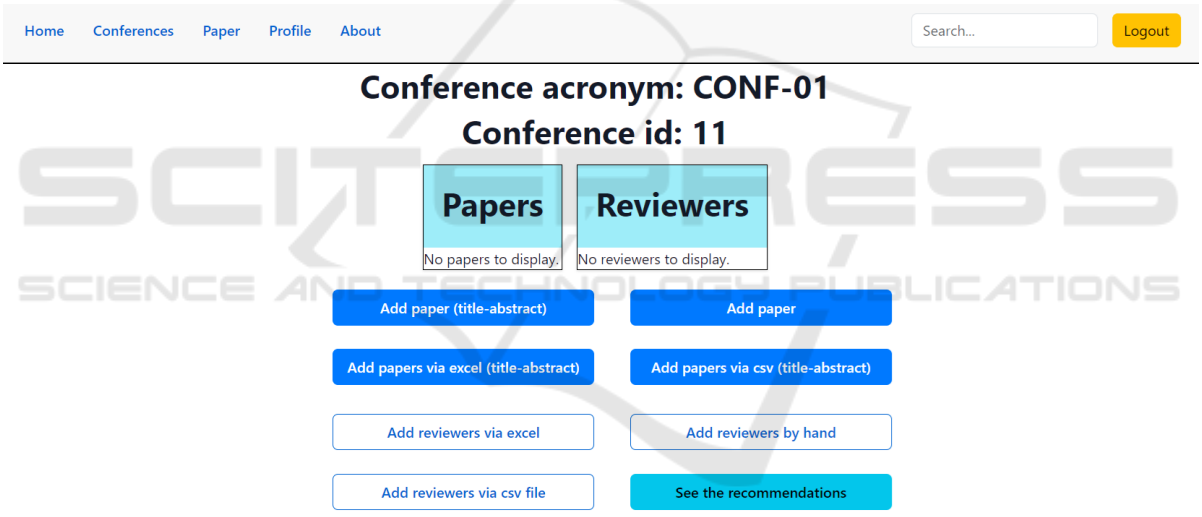

Figure 6: The conference page of the system with all the available options.

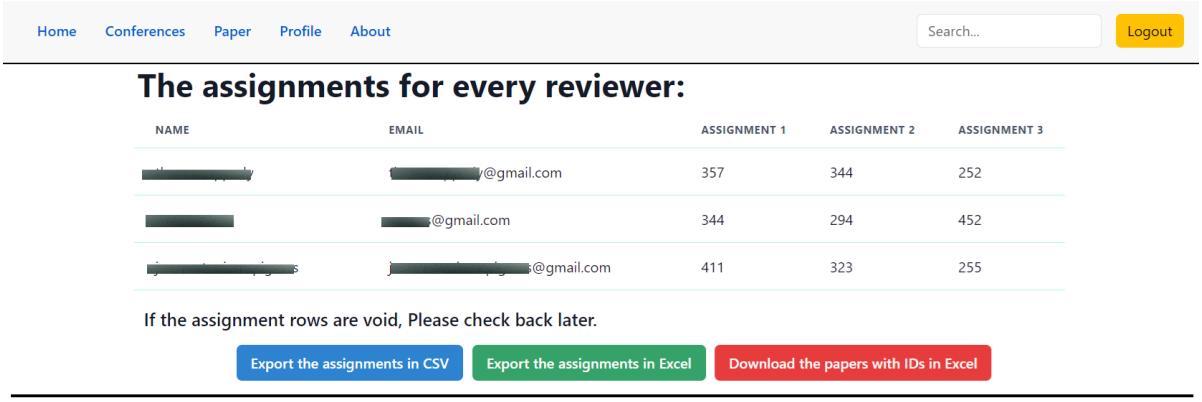

Figure 7: The reviewers-to-papers assignment page of the system.

5 CONCLUSION AND FUTURE PLANS

In conclusion, the process of assigning reviewers to papers is, without question, the most time-consuming task for the organizers of a conference. Therefore, it is absolutely necessary to use an automated system that completes this task and achieves high performance and quality assignments. We strongly believe that a web Conference Management System that solves RAP, is required in order to upgrade a conference quality, by upgrading the peer-review process.

In this work, we propose such a system which exploits the power of LLMs by incorporating them into Recommendation Systems. Regarding our plans for future work:

- We will put our system to online access, in the near future.
- We currently work on the front-end to improve UX-UI issues.
- We will make our system available to all academic personnel organizing a conference.
- We plan to continue running experiments on the performance and efficiency of the reviewer assignment module using different datasets.
- We are going to perform an extensive evaluation and calculate a variety of performance metrics for the reviewer assignment module.

ACKNOWLEDGEMENTS

The work of M. Vassilakopoulos and A. Corral was funded by the Spanish Government MCIN/AEI/10.13039/501100011033 and the "ERDF A way of making Europe" under the Grant PID2021-124124OB-I00.

REFERENCES

Aksoy, M., Yanik, S., and Amasyali, M. F. (2023). Reviewer assignment problem: A systematic review of the literature.

Bouanane, K., Medakene, A. N., Benbelghit, A., and Belhaouari, S. B. (2024). Faircolor: An efficient algorithm for the balanced and fair reviewer assignment problem. *Inf. Proc. & Management*, 61(6):103865.

Charlin, L. and Zemel, R. S. (2013). The toronto paper matching system: An automated paper-reviewer assignment system. In *ICML workshop on peer reviewing and publishing models*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proc. 2019 Conf. of the North American Chapter of the ACL: Human Language Technologies, Volume 1 (Long & Short Papers)*, pages 4171–4186, Minneapolis, MN. ACL.

Latypova, V. (2023). Reviewer assignment decision support in an academic journal based on multicriteria assessment and text mining. In *2023 IX Int. Conf. on Inf. Technology and Nanotechnology (ITNT)*, pages 1–4.

Leyton-Brown, K., Mausam, Nandwani, Y., Zarkoob, H., Cameron, C., Newman, N., and Raghu, D. (2024). Matching papers and reviewers at large conferences. *Artificial Intelligence*, 331:104119.

Maleszka, M., Maleszka, B., Król, D., Hernes, M., Martins, D. M. L., Homann, L., and Vossen, G. (2020). *A Modular Diversity Based Reviewer Recommendation System*, page 550–561. Springer Singapore.

Ribeiro, A. C., Sizo, A., and Reis, L. P. (2023). Investigating the reviewer assignment problem: A systematic literature review. *Journal of Information Science*.

Stelmakh, I., Wieting, J., Neubig, G., and Shah, N. B. (2023). A gold standard dataset for the reviewer assignment problem.

Stergiopoulos, V., Tsianaka, T., and Tousidou, E. (2022). Aminer citation-data preprocessing for recommender systems on scientific publications. In *Proc. 25th Pan-Hellenic Conf. on Informatics*, PCI '21, page 23–27, New York, NY. ACM.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., and Jiang, P. (2019). Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proc. 28th ACM Int. Conf. on Inf. and Knowledge Management*, CIKM '19, page 1441–1450, New York, NY. ACM.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wu, C., Wu, F., Qi, T., and Huang, Y. (2021). Empowering news recommendation with pre-trained language models. In *Proc. 44th Int. ACM SIGIR Conf. on Research and Development in Inf. Retrieval*, SIGIR '21, page 1652–1656, New York, NY. ACM.

Yujian, L. and Bo, L. (2007). A normalized levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1091–1095.

Zhang, Y., Shen, Y., Kang, S., Chen, X., Jin, B., and Han, J. (2024). Chain-of-factors paper-reviewer matching.

Zhao, X. and Zhang, Y. (2022). Reviewer assignment algorithms for peer review automation: A survey. *Inf. Proc. & Management*, 59(5):103028.