Developing a Personalized Study Program Recommender[2]

Marc Scheffler [1], Thilo I. Dieing [1] and Lea Cohausz [1]

Abstract: This paper presents a recommender system designed to match prospective students with study programs in Baden-Württemberg, Germany, streamlining the selection process by providing personalized recommendations based on user queries. Utilizing data from approximately 1,500 study programs and employing natural language processing and machine learning techniques, specifically the German fastText model for word embeddings, our system captures the semantic relationships between user queries and program descriptions. We evaluated the system's performance using both manual test cases and automated validation methods. The manual evaluation involved subjective assessments by multiple raters, while the automated approach utilized self-supervised keyword-based approaches. The results demonstrate the system's effectiveness in enhancing the study program selection process.

Keywords: education, recommender system, study program recommendation, NLP, fastText, embeddings, BERUFENET

1 Introduction

In times when managing the sheer flood of information and retrieving what is relevant is more of a problem than being able to access information and knowledge in the first place, prospective students often struggle to find suitable study programs that align well with their interests and academic goals. This issue is particularly significant considering that academic success is more likely if students' interests, expectations, and abilities fit well with the content of their study program [MKJ23].

Traditional methods of study program selection often involve manual searches through university catalogs, websites, and other informational resources, which can be time-consuming and overwhelming. To address this, we introduce a recommender system that automates and simplifies the process by providing personalized study program suggestions based on users' self-assessment of their interests and skills.

Recommender systems have been extensively researched and implemented in various domains such as e-commerce [HRW21], social-media [An18], and education [UMO21]. Within the field of education, course recommendation is a particularly active area of research, reflected in the numerous studies that have been carried out [Ka24]. Despite this, there has been relatively little focus on developing systems specifically for study programs; only a

1 University of Mannheim, Chair of Artificial Intelligence, 68159 Mannheim, Germany,
marcscheffler@outlook.com, https://orcid.org/0009-0004-1249-3487;
thdieing@mail.uni-mannheim.de, https://orcid.org/0009-0000-2256-0795;
lea.cohausz@uni-mannheim.de, https://orcid.org/0000-0002-6164-3988

2 The article appears in the workshop proceedings of DELFI 2024 and was presented at the EduRS workshop.

few studies have been conducted in the last decade, indicating a significant gap that needs to be addressed [Ka24]. Study program and course recommendations differ considerably in terms of the information available, possible options and, above all, the consequences for the recipients of the recommendations, which underlines the need for further research.

In the context of these educational recommender systems, several approaches have been explored, including content-based filtering, collaborative filtering, and hybrid methods [Sa22]. We consider our recommender system to be located with other content-based filtering approaches, as it uses Natural Language Processing (NLP) and machine learning techniques to recommend study programs based on textual queries. We employ word embeddings to capture the semantic relationships between users' input queries and the text of program descriptions. These are applied to the extensive data we have collected, which is the entire study program offering in Baden-Württemberg.

To test the robustness of our recommender system, we conducted a thorough evaluation using both manual test cases and self-supervised validation methods. The manual evaluation involved subjective assessments from multiple raters, while automated validation utilized keyword extraction techniques and ranking metrics to measure recommendation accuracy.

The recommender is being built as part of the NEST-BW project [NE24] initiated by the Ministry of Science, Research and Art of the State of Baden-Württemberg, Germany. One of the project goals is to create a study orientation guidance system, in which this recommender will ultimately be employed, allowing prospective students to make a better-informed decision when choosing their career path.

2 Methodology

2.1 Data

The Bundesagentur für Arbeit, i.e., the German Federal Employment Agency, is a federal agency providing services for the German labour market. As part of their service, they operate BERUFENET [Bu24], an information portal on the subject of professions in Germany. It contains detailed information on all recognised training occupations and their specialisations as well as on school-based training, *study programs* and further training programs.

Serving as the basis of the recommender system, we scraped all necessary data related to study programs from BERUFENET, yielding a rich dataset of roughly 24,000 study programs in Germany. Entries include program names, detailed descriptions, administrative details, and information about the university itself. For the purpose of this recommender, we limited the data to about 1,500 study programs, which make up the entirety of the state of Baden-Württemberg. Nevertheless, the proposed methodology can easily be adapted to incorporate more study programs.

The system begins with the pre-processing of the text data, i.e., all text we collected related to the study programs, which we refer to below as *program descriptions*. This involves standard NLP techniques. The SoMaJo tokenizer [PU16] is used for tokenizing the German text; the stopwords from both English and German are removed to ensure the relevance of the processed text.

2.2 Word Embeddings

As the embedding model of our choice, we used the free and open-source fastText model [Bo16]. This model creates dense vector representations of words, capturing their meanings based on the contexts they appear in. Since fastText is one of the only word embedding models that provide pre-trained embeddings in the German language and make use of substring information, our choice was straightforward. Its ability to produce vectors for any words, even made-up or misspelled ones, allowed us to focus on the important parts of the system rather than having to deal with more sophisticated preprocessing [Bo16; Eh21].

As an input query to the recommender, we have users provide their interests and abilities in form of a list. An example of such a query would be "Computers, Biology, Soccer, ...". The embeddings obtained through the fastText model are crucial for computing the similarity between query words and program descriptions. We decided to work with word embeddings rather than sentence embeddings since this better allows us to compare the users' input to the tokenized study program descriptions. We do not expect users to articulate their queries in full sentences; hence, we have decided not to use sentence embeddings.

2.3 Recommendation Approach

As the first step, we obtain the word embedding for each word in a given query, resulting in a matrix $\mathbf{Q} \in \mathbb{R}^{m\times 300}$, where m corresponds to the number of words in the query and 300 is the dimensionality of the fastText embeddings.

For each study program in our dataset, we have to quantify how relevant it is to the users' query. To do so, we first compute the cosine similarity between each word embedding in the query $\mathbf{Q}$ and each word in the description of a given study program $\mathbf{P}_i \in \mathbb{R}^{l\times 300}$, where l is the number of words/embeddings in a program's description, varying between programs. We compute $\mathbf{S} = \frac{\mathbf{Q}\cdot\mathbf{P}_i}{\|\mathbf{Q}\|\|\mathbf{P}_i\|}$, resulting in the similarity matrix $\mathbf{S} \in [0,1]^{m\times l}$.

In order to obtain a single real-valued score of how well this study program aligns with the users' query, we have to aggregate this similarity matrix in some way. We proposed several scoring functions and evaluated their results (cf. Abschnitt 3). For this, we came up with three different approaches as well as an additional weighting mechanism to treat certain query words differently. The three scoring functions are *sum*, *mean*, and *mean/sd*, which we'll highlight in the following paragraphs.

The most straightforward approach is to simply *sum* up all the entries in **S**. However, this does introduce a certain bias, as study programs with longer descriptions ($l \gg 1$) implicitly accumulate more points. Hence, a study program unrelated to the query, with exclusively small similarity scores, may still come out as the top match simply due to its length. To alleviate this problem, we introduce a scoring *threshold*, where each individual score below will be set to zero if it is below this threshold. This stops the accumulation of small values and ensures that only relevant words will influence the score.

The second scoring approach is to calculate the *mean* along the first dimension of **S**, obtaining a single mean score $\in [0, 1]$ for each word in the query. The resulting m mean scores can ultimately be summed again to obtain a single real value.

The third approach is to normalize the obtained mean scores for each query word. By dividing the mean scores by their standard deviations, this approach gives more weight to query words with consistent (low-variance) similarity scores. The normalization step reduces the influence of outliers. A high mean score due to a few very high values will have less impact if the standard deviation is also high, making it less sensitive to outliers.

Finally, a last mechanism was implemented to improve the recommendations further. Because of the nature of the scoring approaches considered, a query such as "Program, Data, Computer, Biology" would be biased towards computer science programs, even though programs such as "Bioinformatics" may be better suited given the query. By calculating the cosine similarity between query words, summing their inverses, and normalizing via the softmax function, we obtain intra-query similarity (*IQS*) weights. By employing these IQS weights, the "Biology" query word would receive a higher weight, as it is the most dissimilar from the four.

3 Evaluation

To evaluate the quality of our proposed recommendation system, we developed two approaches. For the first, we manually created 47 test cases where we defined different queries and what an expected result might look like. We selected three configurations of scoring mechanisms that seemed to work well according to our previous tests and manually evaluated the recommender's results. Per the test case, each of the three raters selected one of the three different recommender configurations that they individually perceived to work best. The configurations we chose were:

- Mean*IQS, threshold=0.51
- Sum, threshold=0.51
- Mean, threshold=0.51

To further validate our assessment, we developed a second approach. By using TF-IDF and KeyBERT [Gr20], we were able to extract two distinct sets of keywords from each study program description. These keywords can now be used as a query to the recommender, while at the same time, we had a concrete target study program that should be returned by the recommender as early as possible. As an error metric, we chose the Mean Reciprocal Rank (MRR) to see when, on average, the correct study program appeared in the recommendation list. For example, an MRR score of 0.5 tells us that the correct result appears in position 2 on average, and accordingly, an MRR score of 0.25 indicates that the correct result appears in position 4 on average.

To simulate more realistic queries, we randomly selected the 3-8 most relevant keywords; configurations were chosen in a grid search manner. This approach allows us to evaluate many more configurations much faster than if we proceeded manually.

4 Results

The results of the first evaluation are visualized in Fig. 1. These had an inter-rater agreement of 0.75, indicating a strong agreement between the raters.

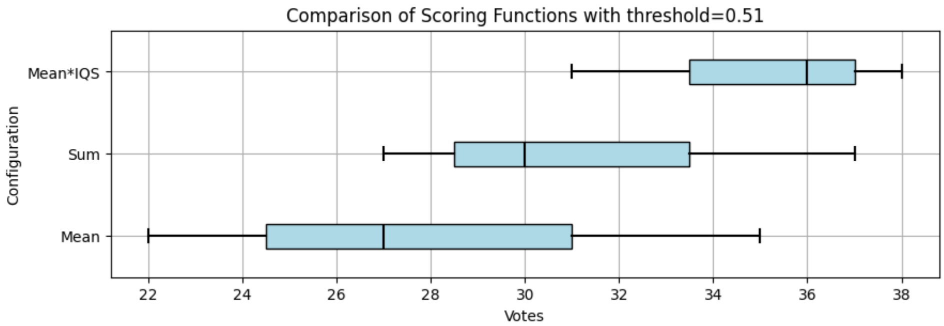

Fig. 1: Number of votes for "best configuration"; 47 test cases, 3 raters' votes combined; selecting more than one configuration was possible.

The third configuration (Mean*IQS) distinctively received more votes than the other two. This result suggests that the IQS was effective in improving the prediction quality.

The second evaluation approach in Fig. 2 shows promising results for various different configurations. Reassuringly so, both KeyBERT and the TF-IDF keyword sets resulted in the same configurations performing well. Our assumption that the IQS mechanism successfully improved the recommendation performance holds for the three specific configurations that were manually tested. On average, however, the configurations without the IQS mechanism are neither better nor worse, regardless of the selected scoring mode.

It is important to note that the automated validation using keyword-based methods does not fully capture the benefits of the IQS mechanism. Weighting the keywords differently won't yield the desired improvements, as these keywords are already well-formed queries to

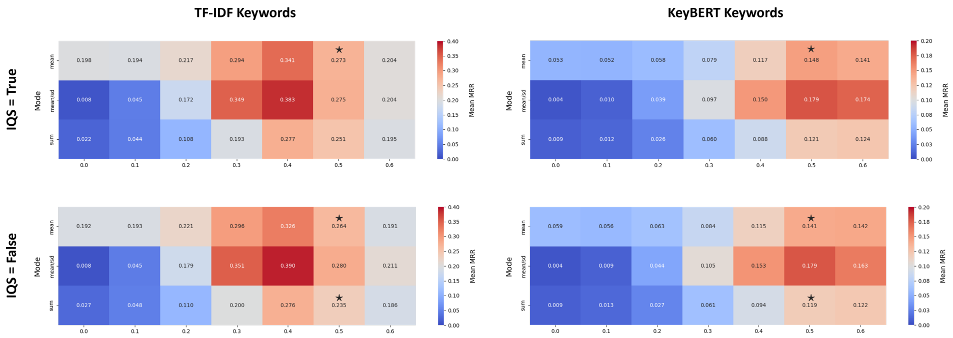

Fig. 2: MRR results for configurations chosen via grid search; each MRR score was computed by evaluating ~1,500 distinct study programs. Configurations highlighted with a star symbol (★) correspond to the ones that tested in Fig. 1.

the target study program. Consequently, the full advantage of the IQS mechanism might be understated by these automated evaluations, which are limited in their ability to mimic human querying behavior.

Ultimately, we see that the normalized mean-scoring mode (”mean/sd”) seems to have performed best out of the three. Unfortunately, we did not consider this scoring mode when performing the manual evaluation; this will be included in a future evaluation. Lastly, while the raters found the plain sum-scoring mode to provide slightly better recommendations than the plain mean-scoring (cf. Fig. 1), these validation results disagree.

5 Conclusion

Navigating the vast amount of information to find suitable study programs is a significant challenge for prospective students. Traditional methods are often inefficient and overwhelming, underscoring the need for automated, personalized recommendation systems. This research presents a recommender system tailored to match students with study programs in Baden-Württemberg, Germany, using publicly available data.

Our evaluation through manual and automated methods clearly demonstrated the effectiveness of our recommender. The scoring configuration using normalized mean scores was particularly effective. These findings demonstrate the system’s potential to assist students in making informed academic decisions. As part of the NEST-BW project, this recommender system aims to improve study orientation guidance. Future work will expand the dataset, refine algorithms, and integrate the system into a practical application to benefit more students with accurate and personalized recommendations. Additionally, we plan to conduct an evaluation to assess how helpful users find the suggestions, enabling continuous improvement based on user feedback and satisfaction.

References

[An18] Anandhan, A.; Shuib, L.; Ismail, M. A.; Mujtaba, G.: Social Media Recommender Systems: Review and Open Research Issues. IEEE Access 6, S. 15608–15628, 2018, DOI: 10.1109/ACCESS.2018.2810062.

[Bo16] Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T.: Enriching Word Vectors with Subword Information. arXiv preprint arXiv:1607.04606, 2016.

[Bu24] Bundesagentur für Arbeit (BA): BERUFENET, 2024, [Online]. Available at: https://web.arbeitsagentur.de/berufenet/,. [Last accessed: 14. Juli 2024].

[Eh21] Ehrmanntraut, A.; Hagen, T.; Konle, L.; Jannidis, F.: Type- and Token-based Word Embeddings in the Digital Humanities. In: Workshop on Computational Humanities Research. 2021, URL: https://api.semanticscholar.org/CorpusID:240290382.

[Gr20] Grootendorst, M.: KeyBERT: Minimal keyword extraction with BERT. Version v0.3.0, 2020, DOI: 10.5281/zenodo.4461265, URL: https://doi.org/10.5281/zenodo.4461265.

[HRW21] Hussien, F. T. A.; Rahma, A. M. S.; Wahab, H. B. A.: Recommendation Systems For E-commerce Systems An Overview. In: Journal of Physics Conference Series. Bd. 1897, 012024. Journal of Physics Conference Series, IOP, 012024, S. 012024, 2021, DOI: 10.1088/1742-6596/1897/1/012024.

[Ka24] Kamal, N.; Sarker, F.; Rahman, A.; Hossain, S.; Mamun, K. A.: Recommender System in Academic Choices of Higher Education: A Systematic Review. IEEE Access 12, S. 35475–35501, 2024, DOI: 10.1109/ACCESS.2024.3368058.

[MKJ23] Messerer, L. A. S.; Karst, K.; Janke, S.: Choose wisely: intrinsic motivation for enrollment is associated with ongoing intrinsic learning motivation, study success and dropout. Studies in Higher Education 48 (1), S. 137–150, 2023, DOI: 10.1080/03075079.2022.2121814, eprint: https://doi.org/10.1080/03075079.2022.2121814, URL: https://doi.org/10.1080/03075079.2022.2121814.

[NE24] NEST-BW Koordinationsstelle: Willkommen bei NEST-BW, 2024, [Online]. Available at: https://www.uni-mannheim.de/nestbw/,. [Last accessed: 12. Juli 2024].

[PU16] Proisl, T.; Uhrig, P.: SoMaJo: State-of-the-art tokenization for German web and social media texts. In (Cook, P.; Evert, S.; Schäfer, R.; Stemle, E., Hrsg.): Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task. Association for Computational Linguistics, Berlin, S. 57–62, 2016, DOI: 10.18653/v1/W16-2607, URL: https://aclanthology.org/W16-2607.

[Sa22] Sandoussi, R.; Hnida, M.; Daoudi, N.; Ajhoun, R.: Systematic Literature Review on Open Educational Resources Recommender Systems. International Journal of Interactive Mobile Technologies (iJIM) 16 (18), pp. 44–77, 2022, DOI: 10.3991/ijim.v16i18.32197, URL: https://online-journals.org/index.php/i-jim/article/view/32197.

[UMO21] Urdaneta-Ponte, M. C.; Mendez-Zorrilla, A.; Oleagordia-Ruiz, I.: Recommendation Systems for Education: Systematic Review. Electronics 10 (14), 2021, ISSN: 2079-9292, URL: https://www.mdpi.com/2079-9292/10/14/1611.