TalentJobRadar: Advanced Data-Driven Recommendations for In-Demand QA Soft Skills and Career Opportunities

Wagner Lancetti
Universidade Federal de São João del Rei
Brazil
Universidade de São Paulo, Brazil
wlancetti@ime.usp.br

Vinicius da Fonseca Vieira
Universidade Federal de São João del Rei
Brazil
vinicius@ufsj.edu.br

Kelly Rosa Braghetto
Universidade de São Paulo, Brazil
kellyrb@ime.usp.br

Vinicius H. S. Durelli
Universidade Federal de São Carlos, Brazil
vinicius.durelli@ufscar.br

Abstract

Introduction: Software Quality Assurance (QA) is essential to ensure that software products meet predefined requirements. While QA tasks are technical, soft skills play a crucial role in project success, product quality, and the productivity of QA professionals. **Objectives:** The main objective of this work is to provide a job and skill recommendation tool focused on the Brazilian QA market. **Methods:** Data was extracted from 2,164 LinkedIn job postings using a data-driven, inductive approach, combining both manual and automated processes. Job descriptions and users' skills were mapped into binary vectors for comparison. The tool displays job recommendations in a card format, showing company name, required skills, LinkedIn links, and the user's suitability for the position. Additionally, it suggests skills for improvement and highlights the top three skills associated with the user's current soft skills, presenting a radar chart that shows job availability by seniority level. **Evaluation:** We conducted a preliminary evaluation of the tool using 45 synthetic profiles representing varying skill levels to simulate diverse user scenarios, allowing us to assess the system's adaptability and effectiveness. Job recommendations demonstrated notable precision, recall, and F1-score values, while skill recommendations showed positive results in terms of relative coverage, precision, and relevance. However, applying filters led to a decrease in overall metrics, highlighting limitations in filtered recommendations. **Conclusion:** The tool shows promise in helping QA professionals at various experience levels identify relevant job opportunities and areas for skill enhancement. Although filtered recommendations present limitations, the system effectively highlights suitable positions and skills for development, supporting employability in the QA field. Future improvements include refining filtered recommendation accuracy and expanding to technical skill recommendations.

Keywords

Quality assurance, QA, Software Testing, Soft skill, Recommender systems, Skill Recommendation

1 Introduction

The software development process is inherently complex, as it involves navigating a wide array of interconnected and multifaceted challenges. The lion's share of this complexity stems from both technical and human factors [10]. On the technical side, software developers must understand and translate complex business requirements into functional software while managing the ever-evolving technological landscape. On the human side, successful software development requires collaboration and communication among developers and other stakeholders.

Software companies tend to prioritize hard skills such as data structures, algorithms, programming, and technical certifications when building teams or hiring new professionals [25]. Over the years, it has become increasingly evident that the success of software projects depends not only on technical or process considerations but also on human factors. Given that people working on software projects must engage and communicate with team members and stakeholders, negotiate with clients, and produce reports, among other non-technical activities, the human dimension is as vital as the technical one [1].

Quality Assurance (QA) involves a mix of automated and manual activities (i.e., non-computer-based testing) [27], so not all QA activities revolve solely around computer-based hard skills. For example, QA professionals must assess the quality of software products from the end users' perspective (e.g., usability testing), take stakeholder feedback into account, and make collaborative decisions. Thus, beyond conventional computer-based testing techniques, soft skills play a crucial role in ensuring the development of high-quality software products.

A survey of over 250 technical leaders cited the primary reason for project failure as a lack of soft skills [6]. When non-technical skills are developed to complement technical skills, personal productivity, collaboration, and synergy increase. This results in higher project success rates, sustainable competitive advantage, and greater profitability.

This research focuses on streamlining the job search on LinkedIn by identifying the most suitable job opportunities based on the user's skills and specified filters. Additionally, for those looking to expand their opportunities, the developed tool suggests which skills should be developed to improve the potential to achieve professional goals. To this end, our research entailed developing a data analysis algorithm that maps the user's skills and identifies the most suitable job opportunities in the QA field. Additionally, a system was implemented to recommend new skills for enhancement based on the user's soft skills and specified filters, maximizing opportunities in the job market. Customized filters were also created to enable users to refine their searches and identify the opportunities that best align with their professional goals.

This work makes a significant contribution to the field of data analysis and recommendation tool development by creating an innovative solution that focuses exclusively on soft skills. The developed tool introduces a novel methodology for evaluating and recommending these often-overlooked competencies, providing personalized job recommendations and soft skill development suggestions aligned with users' professional goals. The originality of this work lies precisely in its unique and innovative approach, focusing solely on soft skills, moving beyond existing solutions that typically prioritize technical skills. This approach introduces a new model for supporting soft skill development, directly impacting users' employability and career advancement.

The remainder of this paper is organized as follows. Section 2 provides background information on Quality Assurance and the role of soft skills in the software industry. Section 3 reviews related work on job recommendation systems and the importance of soft skills in employability. Section 4 outlines the methodology employed in the development of our job recommendation system, providing an account of the data extraction process and the design and implementation of the TalentJobRadar tool. Section 5 presents the main functionalities of the proposed tool, including job and skill recommendations. Section 6 reports the results of a preliminary evaluation, discussing key performance metrics and identifying potential threats to validity. Finally, Section 7 concludes the paper by summarizing the main findings and outlining directions for future work.

2 Background

Quality Assurance (QA) is often used in a more restricted sense, either to denote adherence to a minimum standard or to assure stakeholders that quality is being maintained [37]. In this context, QA encompasses two fundamental principles: "fit for purpose" (the product should meet the intended purpose) and "right the first time" (errors should be eliminated). Additionally, QA involves managing the quality of raw materials, assemblies, products, components, and services related to production, as well as overseeing management, production, and inspection processes [34]. QA includes both automated and manual activities (such as non-computer-based testing) [27], which means it is not limited to technical skills, or hard skills—those linked to specialized knowledge for task execution [30]—but also encompasses interpersonal skills, or soft skills.

Soft skills are essential, yet their exact definition can be somewhat nebulous [24]. Given the imprecision of these definitions and the difficulty in reaching a unified understanding, many authors opt to exemplify these competencies [26]. One study describes soft skills as "non-technical skills that do not rely on abstract reasoning, involving interpersonal and intrapersonal competencies to facilitate performance in specific contexts" [20]. Another definition states that they are "intangible, personality-specific skills that determine one's effectiveness as a leader, listener, negotiator, or conflict mediator," with examples like communication ease, personal habits, and optimism [4]. While the definition of soft skills is broad and varied in the literature, there is consensus on their importance, with several authors asserting that hard skills alone are insufficient to ensure success in many roles [29].

To explore and analyze these competencies on a large scale, specialized tools are required. SkillNER is a tool developed to assist researchers and professionals in identifying and analyzing soft skills in texts using Natural Language Processing (NLP) techniques [14]. Applied to data extracted from LinkedIn, the tool enables the identification and classification of soft skills present in job descriptions, providing a detailed view of the competencies most in demand in the market.

With the identification of these skills, it becomes possible to develop personalized recommendation systems. These systems are essential tools for managing information overload by filtering relevant data and capturing users' preferences, interests, or behaviors [15]. Recommender Systems (RSs) play a crucial role in various information systems, assisting in decision-making and driving business growth [39]. In the recruitment context, RSs are designed to both suggest suitable job openings to candidates based on their qualifications and preferences, and recommend candidates to recruiters based on the specific requirements of the positions [19].

There are two widely used types of recommendation systems: content-based and collaborative filtering. In content-based systems, recommendations are made based on the user's profile characteristics and item attributes [2]. The user's profile is automatically built through monitoring their interactions, and the items are described by specific attributes [11]. User preferences, often represented as "ratings", can be binary or on a scale [3], reflecting their level of interest in certain items.

On the other hand, collaborative filtering recommends items to the user based on the preferences of a group of people with similar profiles. The fundamental principle is to identify users or items that share similar characteristics to predict what the target user will appreciate, even if they have not interacted with the item yet [13]. This approach is widely adopted due to its ability to learn from the collective preferences of a community [33].

To analyze data and develop recommendation systems, various metrics are used to measure association and similarity between variables. The Cramér's V metric is a popular association measure for categorical variables, with estimators of this coefficient being simple functions of Pearson's chi-square statistic, used to assess the strength of association between categorical variables in a contingency table [8]. It is particularly useful for identifying correlations in datasets involving discrete categories.

Cosine similarity, as shown in Equation 1, is a widely used metric for quantifying the similarity between two vectors, $\mathbf{A}$ and $\mathbf{B}$, based on the distribution of their components [17]. Since this metric evaluates similarity in orientation, rather than the size of the vectors [23], it stands out by focusing on the common elements between the vectors, i.e., the values that are non-zero in both vectors. Thus, cosine similarity identifies points of intersection (or overlap) in the directions of the vectors, ignoring differences in magnitude or the number of features present, making it particularly useful when skill vectors have large differences in quantities.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \quad (1)$$

3 Related Work

Several studies have addressed the importance of soft skills and the development of recommendation systems in the context of software engineering and IT. In [25], the author analyzed the most sought-after soft skills in the field of software engineering in Uruguay in 2013, establishing a ranking of the main interpersonal skills based on their recurrence in job descriptions. The study highlights the growing appreciation of these competencies due to their relevance in communication, teamwork, and problem-solving in software project.

In the context of recommendation systems, [5] used Bayesian networks to suggest soft skills in IT job postings, considering both technical and social skills. Two models were created: M*sme*, based on interviews with company representatives, and M*adv*, based on data from job advertisements. The M*adv* model proved to be more effective for candidates, while M*sme* is more suitable for recruiters.

In general recommendation systems, [22] developed MOVREC, using collaborative filtering and the k-means algorithm for movie recommendations based on attributes provided by users, an approach similar to skill recommendation. In turn, [38] proposed a hybrid recommendation system that combines collaborative filtering and content-based filtering to suggest jobs, exploring relationships between job postings and users. This integration aims to increase accuracy by merging similar preferences and textual information from job listings, which, as indicated by [7], can lead to more robust recommendation systems.

4 Methodology

This research combines a quantitative and qualitative exploratory approach, aiming to investigate the most in-demand soft skills for QA professionals in the Brazilian software industry. This initial analysis aimed to identify the key interpersonal skills valued in the market [1], and from this understanding, the need to develop a recommendation tool focused on these competencies emerged.

The study follows a positivist approach, utilizing the collection and analysis of quantitative data to map the recurrence and demand for soft skills. Complementarily, a qualitative analysis was conducted to explore market trends and needs. The methodology adopted was a case study, using job description data extracted from LinkedIn. The data collection involved Natural Language Processing (NLP) techniques for the automatic extraction of skills, along with a manual analysis of job descriptions to validate the identification of interpersonal competencies.

4.1 Data Extraction Method

The first step of this work was data collection, considering job advertisements as source. For this, we created a Python script to extract job listing information from LinkedIn into a spreadsheet, which was later converted into a CSV file. Specifically, our script uses a Tag filter to optimize the search process. We conducted a search using the following tags (provided in both Portuguese and English): QA, QA Tester, Tester, Automation Tester, *Analista de Garantia de Qualidade de Software*, and *Analista de Qualidade de Software*. Since the focus of the work was to obtain an overview of the latest trends in the QA job market in Brazil, the collection was focused on job listings published from March 14, 2023, to March 14, 2024. The data collection occurred between March 14 and 15, 2024. We manually filtered out listings from companies or recruitment agencies whose primary business was not software. The initial data collection resulted in 2,209 job listings. After the initial extraction, we also removed duplicate listings, resulting in 2,178 entries. Additionally, we filtered out listings not related to QA, resulting in a total of 2,164 job listings after the data collection and filtering process.

This significant dataset, consisting of 2,164 job listings, allowed for the formulation of the idea to develop a recommendation system focused exclusively on soft skills. Compared to other studies that utilize considerably smaller datasets—such as 101 [12], 190 [36], 200 [18], and 1,000 [21] job listings—the volume of data in this study strengthens the relevance of creating a recommendation system capable of providing more robust and accurate results based on these skills.

The data extraction process was conducted through the manual reading and analysis of job advertisements to identify relevant information. Since the job listings did not follow a standardized format that facilitated automated extraction, all were analyzed by two researchers, one of whom was an expert in Software Engineering. During the extraction of soft skills, the SkillNER tool — developed to assist researchers and professionals in analyzing interpersonal skills [14] — was used, which facilitated the identification and extraction of these competencies from the selected job ads. The results generated by the tool significantly accelerated the analysis process, ensuring more efficient extraction of soft skills from the advertisements.

The data extracted from each job posting, which also served as filters in the recommendations, were categorized as follows:

- **Soft skill:** Soft skills were only extracted if explicitly requested in a job posting. For example, "ability to work well in a team and collaborate with colleagues to achieve common goals" would be counted as a teamwork soft skill; "our company is all about team collaboration and innovating together" would not.
- **Seniority level:** The seniority levels were categorized as entry-level (i.e., junior), mid-level, and senior based on the terms mentioned in the job postings. Positions mentioned as leader, manager, and director were categorized as senior roles. Similarly, junior or graduate positions were categorized as entry-level roles. Some job postings included an intermediate level between two categories (e.g., entry-mid level); for these cases, the approach adopted was to classify them under the lower of the two levels (i.e., for the entry-mid level example, it would be categorized as entry-level).
- **Job type:** The job postings were classified as either in-person or remote, based on the field provided by LinkedIn or according to the job description when this information was not explicitly stated.

4.2 Tool Development Process

The TalentJobRadar tool developed in this work aims to optimize the job search in the software quality field. Using Streamlit [35] for the interface and Python [31] for the functionalities, it serves as a recommendation system for both jobs and skills tailored to users.

The main functionality of the tool is to allow users to input their interpersonal skills (soft skills) and configure customized filters to refine the results. Based on this information, the tool recommends job opportunities that best align with the user's competencies or suggests skills that would be beneficial to develop in order to increase employability chances.

4.2.1 *Mapping of Skills.*

Based on the analysis of the extracted data, 32 soft skills were identified, which needed to be mapped for the construction of the user profile vectors. Each position in the vector represents a skill, and since job listings do not specify the required proficiency level for each competency, a binary categorization was adopted: if the user or the job listing possesses the skill, the value assigned is 1; otherwise, it is 0.

With a dataset of 2,164 job vacancies, a matrix was constructed that maps all these opportunities, checking for each job description which skills are mentioned. The use of these skill vectors, both for jobs and users, makes the similarity calculations faster and more efficient.

To ensure efficiency and avoid system overload, this matrix was pre-calculated and stored using an auxiliary program, utilizing the zipfile library. This way, the tool only loads these precomputed data during execution, ensuring a fast and high-performance recommendation process.

4.2.2 *Clustering of Skills.*

To enhance the skill recommendations, the competencies were grouped based on the associations found in job demands. Using the Cramér's V calculation, an appropriate metric for evaluating the association between categorical variables, it was possible to build an association matrix between all the skills. This 32x32 matrix maps the correlation between different skills, allowing the identification of which competencies are most frequently related to each other in job advertisements.

Once the association matrix was created, the next step was to group the skills into clusters. The technique used for this grouping was hierarchical clustering, which organizes the data into a binary tree structure. This process begins with each skill as a separate unit in the leaves of the tree, merging them pairwise based on similarity, until all elements are grouped at the root [28]. It is an unsupervised learning technique [28].

After creating the hierarchical clustering tree, it was necessary to determine a cutoff point to form the final skill clusters. This cutoff was adjusted through successive partitions until an ideal segmentation was reached. To evaluate the quality of the formed clusters, the Silhouette score metric was used, which measures the consistency of assigning a data point to a specific cluster by evaluating both the separation between clusters and internal cohesion [9]. Negative values indicate incorrect assignment, while positive values signal a good fit of the data to the cluster [9].

The formation of these groups depends on the filters applied by the user to the database, as the exclusion of certain information directly impacts the presence of skills. Therefore, the data cannot be pre-processed or stored in advance. In other words, every time the user requests skill optimization, the clusters must be recalculated to ensure accurate recommendations.

5 TalentJobRadar

The interface of the recommendation tool proposed in this work was developed using Streamlit, as this library provides a user-friendly interface that allows users to input data, run models, and evaluate results, simplifying the development and deployment process of models [32].

As shown in Figure 1 the user has two search options: one for recommending jobs, detailed in Subsection 5.1, and another for improving skills, which will be discussed in Subsection 5.2. In the side menu, the user can input their soft skills by selecting from a list of 32 skills that may be required by jobs. Additionally, the user can filter their search by seniority and job type (remote or on-site), further customizing the search experience.

5.1 Job Recommendation

The job recommendation is based on the skills that the user has entered into the system and the skills required by the jobs. As described in Subsection 4.2.1, the user's skills are mapped into a binary vector. Then, the similarity between this vector and all the jobs in the similarity matrix is calculated using cosine similarity, and the results are stored in a list. The values are then organized in descending order to create a similarity ranking.

Since the user has the option to apply filters such as seniority and remote work, before recommending the best options — that is, those with the highest similarity — the system first filters the results to retain only the data that meets the user's preferences.

Before recommending a job, the system also calculates which skills would be beneficial for the user to develop in order to improve their suitability for the position. This recommendation takes into account both the user's current skills and the skills required to meet the job's demands. It is important to note that this skill development recommendation does not guarantee 100% suitability for the user, as it considers both the user's existing skills and the job's requirements. For this step, cosine similarity is also used in the new calculations.

As shown in Figure 2, the system presents a set of cards with recommended jobs based on the search criteria. Each card displays the job title, hiring company, required competencies, seniority level, type of work (remote or on-site), a direct link to the LinkedIn job posting, and the user's suitability for the position. Additionally, when relevant, skills that the user could develop to increase their suitability for the job are highlighted in green.

5.2 Skill Recommendation

The second functionality of the system is to present the user with skills that would be beneficial to develop in order to improve their employability in the QA field. The primary goal of this recommendation is to identify skills that complement those the user already possesses, in order to enhance their search for new job opportunities.

The first step of this recommendation is to assess the filters selected by the user. Since the filters directly affect the frequency of skills by removing jobs that do not meet the criteria and, consequently, the associated skills, each application of a filter alters the entire recommendation. Therefore, the calculation described in 4.2.2 needs to be performed again, reconstructing the clusters

Figure 1: Initial screen of the TalentJobRadar recommendation tool

Figure 2: TalentJobRadar job recommendation screen

after removing the entries that did not pass the seniority or job type filters.

The second step of the recommendation is to identify which cluster the user best fits into, considering their skills. To do this, it is necessary to calculate the similarity between the user's skills and those present in the clusters. As with the job recommendation, cosine similarity is used for this operation.

Once the cluster that best matches the user's skills is defined, the frequency of these competencies in the job listings is calculated, creating a ranking of the most demanded skills. Based on this ranking, the most frequent skills that the user does not yet possess are recommended.

In some cases, the recommendation from the most similar cluster may return only one or two skills. To address this issue and ensure that three recommendations are made, the system uses the second most similar cluster and recalculates the frequency of the skills. Fewer than three recommendations are returned only if the user already possesses 30 or more skills.

To improve the visualization of the impact of the recommended skills, the system generates a radar chart displaying the number of recommendations for each level of seniority. This chart is based on the user's skill vector, the recommended skills added to this vector, and the already filtered data. In this way, the user is shown the number of new opportunities that would arise if the recommended skill were included in their resume.

The Figure 3 displays the results of the skills recommendation, showing the frequency of each skill within the filtered data. Additionally, a radar chart provides an overview of job demand by seniority level, illustrating the distribution of positions that require these skills across different seniority categories.

6 Preliminary Evaluation

The primary objective of this work is to provide a methodology and tool that enhance the alignment between candidates and job opportunities, with a particular emphasis on soft skills relevant to the QA field. Rather than adopting an approach centered on testing methods with labeled datasets, this research focuses on an analysis based on varied synthetic user profiles, allowing us to preliminarily evaluate the system's adaptability and performance across diverse scenarios.

To validate the proposed tool—which includes two recommendation systems—a preliminary evaluation was conducted using 45 synthetic user profiles randomly generated to represent a range of experience levels within QA. These profiles were divided into three groups to simulate a diverse user base, from early-career professionals to those with advanced expertise. Specifically, the groups included 15 profiles with few skills (3 to 6), 15 with intermediate skills (10 to 15), and 15 with extensive skills (20 to 32). This distribution enabled a comprehensive evaluation of the system's effectiveness across profiles with varying degrees of competency.

The system allows the application of filters based on seniority (entry-level, intermediate, and senior) and job type (remote or onsite). For each profile, all 32 possible filter combinations were tested, in addition to the analysis without filters, meaning considering all available job opportunities for recommendation.

The evaluation metrics used were: precision, recall, and F1-Score for job recommendations (2, 3, 4), and relative coverage, relative precision, and relative relevance for skill recommendations (5, 6, 7). Precision quantifies the proportion of relevant jobs among the 10 recommendations provided by the system, while recall assesses the system's ability to identify relevant jobs across the entire dataset. The F1-Score balances precision and recall, offering a holistic measure of system performance. Relative coverage captures the variation in the number of relevant jobs before and after skill recommendations, considering the entire dataset. Relative precision evaluates the number of relevant jobs among the top 10 recommended after skill recommendations. Relative relevance reflects the increase in the proportion of relevant jobs relative to the total available after skill recommendations. A job was considered relevant if at least 50% of the user's skills matched the skills required for the job.

The following abbreviations are used throughout the equations:

- RelJob = Number of Relevant Jobs
- RecJob (10) = Recommended Jobs (Top 10)
- TotalRelJob = Number of Total Relevant Jobs in Dataset
- RelJobAS = Relevant Jobs, in entire dataset, after Recommended Skills
- RelJobBS = Relevant Jobs, in entire dataset, before Recommended Skills
- RelJobAS10 = Relevant Jobs, top 10 recommended, after Recommended Skills
- RelJobBS10 = Relevant Jobs, top 10 recommended, before Recommended Skills

$$\text{Precision} = \frac{|RelJob|}{|\text{RecJob (10)}|} \tag{2}$$

$$\text{Recall} = \frac{|RelJob|}{|\text{TotalRelJob}|} \tag{3}$$

$$\text{F1\_score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

$$\text{Relative Coverage(\%)} = |\text{RelJobAS}| - |\text{RelJobBS}| \tag{5}$$

$$\text{Relative Precision(\%)} = \frac{|\text{RelJobAS10}| - |\text{RelJobBS10}|}{|\text{RelJobBS10}|} * 100 \tag{6}$$

$$\text{Relative Relevance(\%)} = \frac{|\text{RelJobAS}| - |\text{RelJobBS}|}{|\text{RelJobBS}|} * 100 \tag{7}$$

When users were tested without filters, the system showed robust performance, especially in profiles with multiple skills, as illustrated in Figure 4. The high accuracy for entry-level profiles indicates that the tool can identify suitable job vacancies, even for users with few skills, successfully matching their profiles with corresponding openings. However, for profiles with intermediate skills, performance tends to be less efficient since the user may not have the most commonly required skills in job offers, leading to less accurate matches.

This improves significantly for senior profiles, which have a broader and more diverse skill set, with many skills being more frequent in jobs. This is because the similarity metric used, based on cosine similarity, focuses more on common features between the user profile and job openings, rather than considering differences, optimizing matches for more experienced profiles.

For intermediate profiles, the performance drop is due to the fact that, while profiles with few skills tend to make better matches with vacancies that don't require the most common skills, intermediate profiles may have many skills but not necessarily the most frequently requested ones. As a result, the match is impaired because the system fails to find the most common skills in the vectors, reducing match accuracy.

In the scenarios where no restrictive filters were applied to the presented data, the system achieved a precision of 68.44%, meaning that over 68% of the recommended job postings were relevant. The recall of 47.54% suggests that the system was able to identify nearly half of the relevant opportunities available, while the F1-Score of 33.45% represents a good balance between precision and recall. The relative coverage of 6.63% indicates that the system was able to add new relevant job postings to the database, while the relative precision of 13.08% shows that, when focusing on the top 10 recommended positions, a significant portion of them were relevant. The relative relevance of 16.14% indicates a significant improvement

Figure 3: TalentJobRadar skills recommendation screen

in the quality of the recommendations, suggesting that the system enhanced the relevance of the recommended positions in relation to the total number of available opportunities.

When filters were applied, the results showed a decrease in system performance. The precision for the total set of 45 profiles was 16.44%, significantly lower than in the filter-free scenario. Recall (5.16%) and F1-Score (6.00%) also showed considerable reductions, suggesting that the application of restrictive filters hindered the system's ability to identify relevant opportunities. The relative coverage was -0.12%, indicating that by applying filters, there was no substantial improvement in the number of relevant vacancies after the skills recommendation, and there may have even been a slight decrease in identified opportunities. The relative precision (0.61%) and relative relevance (1.86%) also showed limited impact, suggesting that although the system attempted to focus on more specific jobs, overall performance was not optimized, leading to less effective identification of suitable opportunities.

Figure 5 provides a comprehensive summary of the metrics obtained by the tool. The results indicate that the tool is effective for profiles with both many skills and few skills, but for profiles with an intermediate amount of skills, its performance is not as good as in the other scenarios, as discussed above. However, the tool faces limitations when restrictive filters are used. The performance drop with the application of filters suggests that the system misses relevant opportunities by restricting options, highlighting the importance of balancing filters and flexibility. Furthermore, the relative relevance demonstrates that the system has the potential to improve the quality of recommendations, especially when considering the evolution of users' skills over time.

6.1 Threats to Validity

One potential threat to validity in our study concerns the temporal scope of data collection. Our analysis is based on job postings from a specific period, which limits our ability to capture the evolution of soft skills requirements over time. Consequently, our findings may not fully reflect long-term trends or shifts in industry demands. Future research could address this limitation by incorporating longitudinal data to observe how soft skill requirements change over time. This would not only provide insights into evolving industry demands but also enable the continuous refinement of our model, ensuring that it remains aligned with current trends and accurately reflects the dynamic nature of the job market.

As pointed out in [16], job advertisements may be written by human resources departments, which may reuse templates without modifying them to match the specific soft skill requirements of the advertised position. Consequently, the soft skills listed in these job postings do not always accurately reflect the actual needs of the role, and essential skills may be omitted from the advertisements. To mitigate this issue, a data extraction process was followed that included both manual (i.e., two researchers reviewed the job postings and discussed the results of the automated analysis to ensure a more accurate understanding and extraction of the relevant soft skills) and automated steps.

The main limitation of the tool is the handling of static data. This means that the recommended data is not updated in real-time.

7 Concluding Remarks

This study presented and conducted a preliminary evaluation of a job and skill recommendation tool focused on the Software Quality

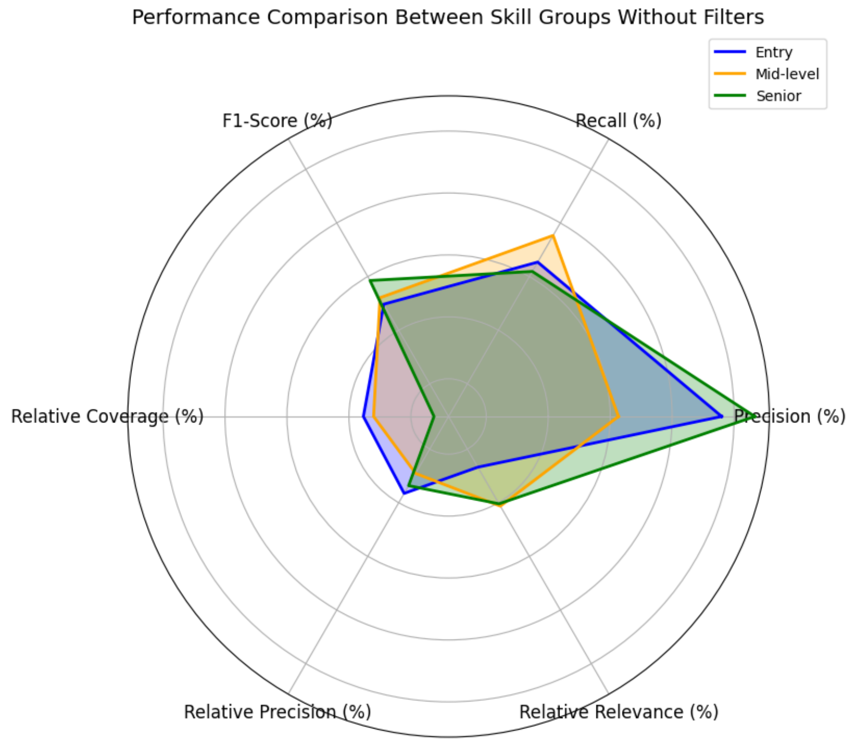

Figure 4: Performance comparison among skill groups without filters

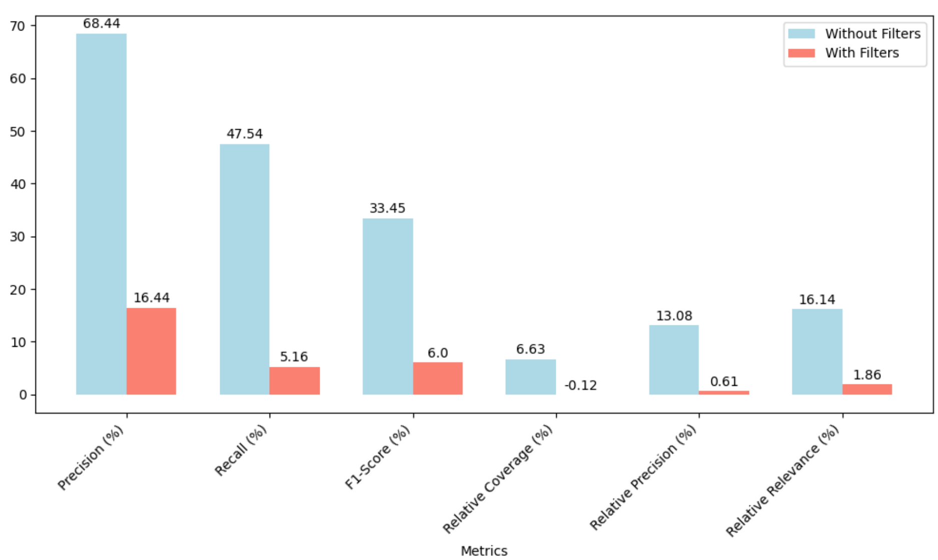

Figure 5: Performance comparison (with and without filters)

Assurance (QA) field, designed to cater to users with varying levels of experience. Using metrics such as precision, recall, F1-Score, relative coverage, relative precision, and relative relevance, the system was tested across different skill profiles and with the application of configurable filters for job types and seniority levels.

The results would seem to indicate that the tool achieves reliable performance for user profiles characterized by either an extensive or very limited skill set, particularly in scenarios where no seniority level or job type filters are enforced. The high precision and recall for these profiles indicate that the system is effective in identifying relevant job opportunities for both users with a diverse skill set and those with only a few competencies, as long as no additional restrictions are imposed.

In contrast, profiles with an intermediate number of skills do not exhibit the same level of performance, suggesting that the distribution of skills may influence the system's effectiveness. Additionally, while the application of filters, while useful for personalizing recommendations, led to a significant drop in overall performance, suggesting that the system may be missing relevant opportunities by restricting the scope of recommendations.

The results suggest the need for adjustments in the system to better identify suitable opportunities, especially in scenarios where recall was reduced. The relative relevance suggests a positive potential in adapting recommendations as the user's skills develop, but it highlights the importance of balancing personalization with the breadth of opportunities, considering the limitation of providing only 10 recommendations.

Future research could explore techniques to balance precision and recall when specific filters are applied, enhancing both the flexibility and usability of the system. Expanding the system to other fields beyond QA could validate its applicability in different professional contexts. As extensions of this work, it is proposed to develop a version of the system that utilizes real-time updated data, allowing for continuous and agile market analysis. Additionally, expanding the system's scope to include technical (hard) skills would integrate technical requirements with interpersonal skills, providing a more comprehensive and relevant view of job market demands. With this expanded approach, users would be able to search for and develop skills more effectively, maximizing their employability and successfully entering the QA job market.

In addition, the results from the applied metrics underscore the necessity of evaluating the platform in collaboration with domain experts. Such an assessment would facilitate the identification of potential deficiencies in the filtering algorithms and, if present, enable the exploration of alternative strategies to enhance their effectiveness and precision.

This research contributes to the field of information systems, specifically in the context of recommendation systems aimed at employability. The developed tool emphasizes and promotes the development of interpersonal skills, which are essential for users' professional growth. Additionally, the tool has the potential to adapt to different user profiles, although challenges remain in optimizing its accuracy and suitability, particularly in scenarios with more restrictive criteria.

Acknowledgements

The authors would like to thank the Federal University of São João del Rei for financial support through a scholarship granted to Wagner Lancetti.

References

[1] S.T. Acuna, N. Juristo, and A.M. Moreno. 2006. Emphasizing Human Capabilities in Software Development. IEEE Software 23, 2 (2006), 94–101.

[2] Charu C Aggarwal et al. 2016. *Recommender systems*. Vol. 1. Springer.

[3] Rishabh Ahuja, Arun Solanki, and Anand Nayyar. 2019. Movie recommender system using k-means clustering and k-nearest neighbor. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 263–268.

[4] K Alex. 2009. *Soft skills*. S. Chand Publishing.

[5] Azuraini Abu Bakar and Choo-Yee Ting. 2011. Soft skills recommendation systems for IT jobs: A Bayesian network approach. In *2011 3rd Conference on Data Mining and Optimization (DMO)*. IEEE, 82–87.

[6] Randy Bancino and Claire Zevalkink. 2007. Soft skills: the new curriculum for hard-core technical professionals. *Techniques: Connecting Education and Careers (J1)* 82, 5 (2007), 20–22.

[7] Justin Basilico and Thomas Hofmann. 2004. Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning*. 9.

[8] Wicher Bergsma. 2013. A bias-correction for Cramér's V and Tschuprow's T. *Journal of the Korean Statistical Society* 42, 3 (2013), 323–328.

[9] Giulia Bruno, Tania Cerquitelli, Silvia Chiusano, and Xin Xiao. 2014. A clustering-based approach to analyse examinations for diabetic patients. In *2014 IEEE International Conference on healthcare informatics*. IEEE, 45–50.

[10] Luiz Fernando Capretz and Faheem Ahmed. 2010. Why do we need personality diversity in software engineering? *ACM SIGSOFT Software Engineering Notes* 35, 2 (2010), 1–11.

[11] Oualid Chenni, Yanis Bouda, Hamid Benachour, and Chahnez Zakaria. 2015. A content-based recommendation approach using semantic user profile in e-recruitment. In *Theory and Practice of Natural Computing: Fourth International Conference, TPNC 2015, Mieres, Spain, December 15-16, 2015. Proceedings 4*. Springer, 23–32.

[12] Maya Daneva, Chong Wang, and Patrick Hoener. 2017. What the Job Market Wants from Requirements Engineers? An Empirical Analysis of Online Job Ads from the Netherlands. In *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. ACM, 448–453.

[13] Mehdi Elahi, Francesco Ricci, and Neil Rubens. 2016. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review* 20 (2016), 29–50.

[14] Silvia Fareri, Nicola Melluso, Filippo Chiarello, and Gualtiero Fantoni. 2021. SkillNER: Mining and mapping soft skills from any text. *Expert Systems with Applications* 184 (2021), 115544.

[15] Mauricio Noris Freire and Leandro Nunes de Castro. 2021. e-Recruitment recommender systems: a systematic review. *Knowledge and Information Systems* 63 (2021), 1–20.

[16] Matthias Galster, Antonija Mitrovic, Sanna Malinen, and Jay Holland. 2022. What soft skills does the software industry really want? An exploratory study of software positions in New Zealand. In *Proceedings of the 16th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 272–282.

[17] Dani Gunawan, CA Sembiring, and Mohammad Andri Budiman. 2018. The implementation of cosine similarity to calculate text relevance between two documents. In *Journal of physics: conference series*, Vol. 978. IOP Publishing, 012120.

[18] Andrea Herrmann. 2013. Requirements Engineering in Practice: There Is No Requirements Engineer Position. In *Requirements Engineering: Foundation for Software Quality*. Springer, 347–361.

[19] Wenxing Hong, Siting Zheng, Huan Wang, and Jianchao Shi. 2013. A job recommender system based on user clustering. *J. Comput.* 8, 8 (2013), 1960–1967.

[20] Scott A Hurrell, Dora Scholarios, and Paul Thompson. 2013. More than a 'humpty dumpty'term: Strengthening the conceptualization of soft skills. *Economic and Industrial Democracy* 34, 1 (2013), 161–182.

[21] Mohamad Kassab, Phillip Laplante, Joanna Defranco, Valdemar Vicente Graciano Neto, and Giuseppe Destefanis. 2021. Exploring the profiles of software testing jobs in the United States. *IEEE Access* 9 (2021), 68905–68916.

[22] Manoj Kumar, DK Yadav, Ankur Singh, and Vijay Kr Gupta. 2015. A movie recommender system: Movrec. *International journal of computer applications* 124, 3 (2015).

[23] Suja Cherukullapurath Mana and T Sasipraba. 2021. Research on cosine similarity and pearson correlation based recommendation models. In *Journal of Physics: Conference Series*, Vol. 1770. IOP Publishing, 012014.

[24] Miriam L Matteson, Lorien Anderson, and Cynthia Boyden. 2016. "Soft skills": A phrase in search of meaning. *portal: Libraries and the Academy* 16, 1 (2016), 71–88.

[25] Gerardo Matturro. 2013. Soft skills in software engineering: A study of its demand by software companies in Uruguay. In *2013 6th international workshop on cooperative and human aspects of software engineering (CHASE)*. IEEE, 133–136.

[26] Gerardo Matturro, Florencia Raschetti, and Carina Fontán. 2019. A Systematic Mapping Study on Soft Skills in Software Engineering. *J. Univers. Comput. Sci.* 25, 1 (2019), 16–41.

[27] Glenford J. Myers, Corey Sandler, and Tom Badgett. 2011. The Art of Software Testing. Wiley, 240 pages.

[28] Frank Nielsen and Frank Nielsen. 2016. Hierarchical clustering. *Introduction to HPC with MPI for Data Science* (2016), 195–211.

[29] Martin Orsted. 2000. Software development engineer in Microsoft: a subjective view of soft skills required. In *Proceedings of the 22nd international conference on Software engineering*. 539–540.

[30] Elizabeth Rainsbury, David Leslie Hodges, Noel Burchell, and Mark C Lay. 2002. Ranking workplace competencies: Student and graduate perceptions. (2002).

[31] Luciano Ramalho. 2015. *Fluent Python: Clear, concise, and effective programming.* " O'Reilly Media, Inc.".
[32] Nikhilesh Sarangpure, Vipul Dhamde, Ankita Roge, Janhawi Doye, Shivam Patle, and Sukhad Tamboli. 2023. Automating the machine learning process using PyCaret and Streamlit. In *2023 2nd International Conference for Innovation in Technology (INOCON)*. IEEE, 1–5.
[33] Mr Sridhar Dilip Sondur, Mr Amit P Chigadani, and Shantharam Nayak. 2016. Similarity measures for recommender systems: a comparative study. *Journal for Research* 2, 3 (2016).
[34] Lionel Stebbing. 1993. Quality assurance: the route to efficiency and competitiveness. *(No Title)* (1993).
[35] Streamlit. 2019. Streamlit: The Fastest Way to Build Data Apps. https://streamlit.io. Acessado: 2024-06-25.
[36] Chong Wang, Pengwei Cui, Maya Daneva, and Mohamad Kassab. 2018. Understanding What Industry Wants from Requirements Engineers: An Exploration of RE Jobs in Canada. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. ACM, 10.
[37] David Woodhouse. 2004. The quality of quality assurance agencies. *Quality in Higher Education* 10, 2 (2004), 77–87.
[38] Shuo Yang, Mohammed Korayem, Khalifeh AlJadda, Trey Grainger, and Sriraam Natarajan. 2017. Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach. *Knowledge-Based Systems* 136 (2017), 37–45.
[39] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* 52, 1 (2019), 1–38.