Ingeniería y Universidad
ENGINEERING FOR DEVELOPMENT

Vol. 28
Continuous Publication
2024

ISSN: 2011-2769 (Online) | ISSN: 0123-2126 (Print)

Artículos

A Job Recommender System for the Unique Framework of Classification of Occupations in Colombia (CUOC) Via Collaborative Filtering*

Sistema de recomendación de empleo basado en el marco de Clasificación Única de Ocupaciones para Colombia CUOC utilizando filtros colaborativos

Claudia Marcela Caro Cortés [a]
Universidad Sergio Arboleda, Colombia
ORCID: https://orcid.org/0009-0001-9343-8014

Juan Pablo Ospina López
Universidad Sergio Arboleda, Colombia
ORCID: https://orcid.org/0000-0002-6841-9778

DOI: https://doi.org/10.11144/Javeriana.iued28.jrsu

Received: 03 April 2024
Accepted: 16 may 2024
Published: 20 december 2024

Abstract:

Objective: The objective of this research is to improve the match between labor supply and demand in Colombia by using machine learning techniques and the Unique Framework of Classification of Occupations in Colombia (CUOC). This framework allows us to enhance the alignment between resumes and job offers, helping job seekers obtain personalized job recommendations based on their profiles.
Materials and Methods: This proposal uses a combination of clustering, classification, and collaborative filtering algorithms to obtain the ten best available vacancies for a particular resume. Standardization of resumes and job offers was performed during the preprocessing stage. We utilized natural language processing algorithms to extract attributes from the CUOC framework. For the training process, we initially employed the K-means algorithm to group the attributes of the CUOC framework. Therea fter, we used KNN, DNN, and AdaBoost as classification algorithms to develop a model that best correlates a resume with a group of vacancies. Finally, a web application was developed using the Django framework, providing a user-friendly interface for job seekers to receive recommendations on the basis of the model outcomes.
Results and Discussion: The best model was selected on the basis of accuracy and processing time. The results indicate that the highest accuracy and recommendation performance were achieved via the CUOC framework, generating a recommendation of the top 10 vacancies on the basis of their similarity level.
Conclusion: Including CUOC characteristics in both resumes and vacancies allows for better matching within the context of the Colombian labor market.
Keywords: Job Recommender, Machine Learning, Natural Language Processing, Employment in Colombia.

Resumen:

Objetivo: el objetivo de la presente investigación se centra en mejorar el encuentro entre la oferta y la demanda laboral en Colombia mediante el uso de algoritmos de máquina y el marco de Clasificación Única para Colombia CUOC, lo que permitirá mejorar **el** método de emparejamiento y permitirá a los buscadores obtener una recomendación de vacantes adecuada con su perfil.
Materiales y métodos: se evaluó los resultados usando dos experimentos generales. Se analizaron las hojas de vida y vacantes sin agregar los atributos que aportan la CUOC. Para lograr la estandarización de las hojas de vida y vacantes en la fase de preprocesamiento, y poder extraer nuevos atributos implícitos a partir de la CUOC, se usaron algoritmos de lenguaje natural. Para entrenar gran cantidad de información se propuso el uso de agrupamiento con el algoritmo de K-Means y clasificación de la información con el uso de los algoritmos KNN, DNN y AdaBoost de tal forma que se encontrara el modelo con mejor precisión y mejor tiempo de procesamiento. Este estudio implementa un sistema de recomendación por filtrado colaborativo basado en usuario.
Resultados y discusiones: los resultados obtenidos muestran que la mejor precisión y recomendación se obtuvo mediante el uso de la CUOC.
Conclusión: incluir las características del CUOC tanto en las hojas de vida como en las vacantes permite un mejor emparejamiento dentro del contexto del mercado laboral colombiano.
Palabras clave: recomendador de empleo, aprendizaje de máquina, procesamiento de lenguaje natural, empleo en Colombia.

Author notes

[a] Correspondence author. E-mail: claudia.caro01@usa.edu.co

Introduction

Job recommendation systems are designed to meet various needs depending on the approach taken. They can help job seekers find suitable positions, assist job recruiters in efficiently managing the selection process via recommendation system technologies [4], and provide relevant job suggestions on the basis of the skills and interests of the job seekers [5].

In terms of local needs, there are no employment recommendation systems that operate according to the standards and variations of the Colombian labor market. Employment providers in Colombia are required to have a technological tool for the registration and management of job portals with vacancies [6]. However, they do not currently offer recommendation systems. Instead, they provide human management services for guidance and direction, which are sometimes free. Many people find it challenging to access this type of support because of factors such as time availability, costs, and long waiting periods for interviews to be conducted.

One reason for the ineffectiveness of job portals in the Colombian market is that they match resumes and vacancies on the basis of keywords in vacancy descriptions, titles, or by relating their position to skills from international frameworks such as the International Standard Classification of Occupations 2008 (ISCO-08), the National Classification of Occupations (CNO), or the adaptation for Colombia developed by DANE CIUO-08 A.C. These frameworks do not fully align with the Colombian labor market, leading to gaps or discrepancies over time when working with different frameworks [1]. To address this, the Unique Classification of Occupations for Colombia (CUOC) was created by the National Administrative Department of Statistics of Colombia (DANE) in 2022. It acts as a unique reference for identifying occupations, labor mobility, human talent management, and labor intermediation. It is also used to analyze the job market, produce statistics, and assess the components of the National Qualifications System. A detailed description of the framework can be found in [1].

Figure 1 illustrates the five-level pyramidal hierarchical structure of the CUOC framework. This structure allows for the classification of the entire occupational landscape by grouping occupations on the basis of their similarity in terms of the level of skills and specialization required for a particular job. The classification resulted in the following categories:

- 10 large groups.
- 43 main subgroups.
- 136 subgroups of the main subgroups.
- 449 primary groups from the subgroups of the previous level.
- 676 occupations resulting from the disaggregation within the primary groups.

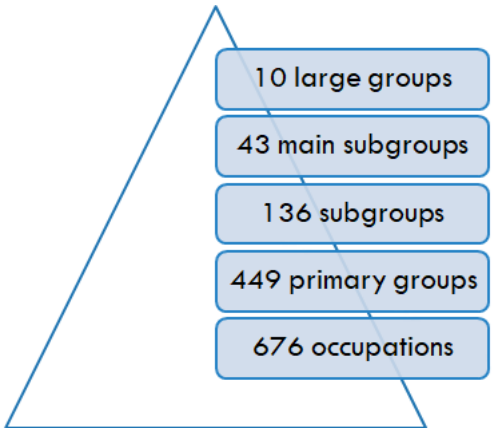

FIGURE 1.
CUOC hierarchical structure classification
Source: Authors' own creation.

Some of the limitations of the current approaches in Colombia, which are based on search systems rather than recommendation systems, are as follows:

- On most search platforms, job seekers must use manual search filters that allow them to search for vacancies they believe may be suitable for their job profile on the basis of their own criteria. This approach does not yield optimal results.
- Job offers displayed to job seekers are typically ordered by publication date, from the most recent to the oldest, which does not ensure better alignment with the job seeker's profile.
- Employment platforms in Colombia yield numerous results, leading job seekers to apply for vacancies where there is a low likelihood of being hired.

The primary motivation for this study is to provide job seekers with the top 10 vacancies that best match their skills and professional background via a recommendation system based on the CUOC framework. A detailed comparison between current job search systems and the proposed system is presented in Table 1.

TABLE 1.
Comparison Between Current Job Search Systems and the Proposed System

Job portals	Proposed system
Diversification and disuse of the reference frameworks used in the positions registered on the current employment platforms.	Standardizes vacancy information in accordance with the Colombian reference framework (CUOC) in its latest version.
The search engine requires manual filtering of vacancies, causing delays in generating search results, and failing to evaluate proper similarity with the candidate's profile.	The match is conducted in real time, providing results within seconds, and allows for a display of vacancies that best align with the candidate's profile.
The match between a resume and a vacancy relies on the number of exact word matches. If the resume does not convey information on skills and aptitudes aligned with work experience, obtaining the most suitable vacancies is not feasible.	The system extracts key attributes of both the vacancy and the job profile using the CUOC, broadening the search to include attributes not explicitly stated in the resume or vacancy text.

Source: Authors' own creation.

Related work

In the classification approach for the development of job search systems, a wide range of algorithms have been used. Among them are DEEP-LAJAM, a continuous learning model employed for matching in a collaborative filtering system [7]; linear support vector machines (Linear SVM) evaluated to optimize vacancy ranking [8, 9]; K-nearest neighbors (KNN), applied in online recommendations without user history [10], for predicting fraudulent vacancies [11] and for resuming selection by comparison with a large volume of vacancies [12]; extreme gradient boosting (XGBoost) for classifying textual documents [13, 14] and predicting candidate suitability with a maximum average precision of 95.14 %; random forests (Random Forest) to predict employment recommendations on the basis of job seekers' preferences and the geographic location of companies [15]; and Bayesian decision trees for candidate selection [16].

In the development of search systems based on neural networks, the models that yielded the best results were recurrent neural networks (RNNs) utilizing historical navigation data with input sequences represented semantically at the word level [17, 18]; long short-term memory (LSTM) networks for discovering knowledge relationships [19]; long sequences for resolving training losses [20]; the hierarchical resume segmentation methodology for learning text representations [21]; implicit candidate intentions or recruiter

refinements obtained in the recommendation process [22]; and bidirectional long short-term memory (Bi-LSTM) networks for learning aggregated representations to generate competencies from skills [23].

Similarly, convolutional neural networks (CNNs) have been used for text processing [23]. Deep Q-learning neural networks (DQNs) have been utilized for skill recommendation, predicting high-level skills from resumes [24]. Deep neural networks (DNNs) are employed to analyze different domains from various data sources, and also to transform multivariate numerical time series of clicks into temporal symbolic sequences [25]. These networks also generate a distributed representation of each job on the basis of categorical attributes [26], in addition, they capture context attributes, and manage large volumes of data where compression and reduced processing time are essential [27]. Finally, text convolutional neural networks (TextCNNs) have been applied to study job predictions [28].

On the other hand, the two-way model of Bi-Deep Factoring Machines (biDeepFM) integrates a factorization machine (FM) to explicitly learn candidate feature interactions alongside dense neural networks for higher-order interactions within an employment recommendation system [5]. Additionally, the deep structured semantic model (DSSM) has been employed for processing large volumes of information, utilizing a semantic representation of sparse data [20].

Finally, we can observe the clustering approach used for developing job search systems. The models that yielded the best results included a K-means model for grouping similar documents into subgroups to reduce the search space, combined with term frequency-inverse document frequency (TF-IDF) to obtain weights [29]. The model was evaluated via a word frequency vector [30, 31], along with hybrid deep collaborative filtering (HDCF), which works with limited or cold data through deep learning [32]. Additionally, cosine similarity, collaborative filters, the Pearson correlation coefficient, and the Tanimoto coefficient were employed to create a hybrid recommendation system that addresses the cold start problem. This approach prioritizes information quality through selective tracking [20].

Materials and methods.

Materials

This study uses data from resumes and applications provided by the Public Employment Service of Colombia (SPE). Vacancy information was extracted through web scraping from the employment exchange of the SPE. The entire dataset comprises 1,700,000 records of resumes, 250,000 records of current vacancies, and 800,000 records of applications. The attributes of the training dataset related to the applications are presented in Table 2.

TABLE 2.
Complete list of application dataset attributes

Attribute	Type	Description
job_seeker_state	object	State where the person who registered their resume resides.
job_seeker_city	object	City where the person who registered their resume resides.
age	int64	Age of the person who registered their resume at the time the information was obtained.
study_level	object	Highest educational level obtained by the person who registered their resume.
last_study	object	Last academic title obtained by the person who registered their resume.
professional_profile	object	Professional profile written by the job seeker who registered their resume.
training	object	Training details provided by the job seeker who registered their resume.
job_title	object	Title of the vacancy provided by the employer.
job_description	object	Description of the activities and requirements of the vacancy, provided by the employer.
job_requied_studies	object	Highest level of education required to fill the vacancy, as specified by the employer.
job_discipline_profession	object	Profession or qualification required by the employer, applicable to technical and professional studies.
job_entry_salary	object	Salary range specified by the employer.

work_position	object	Position for the vacancy, provided by the employer.
job_state	object	State where the activities related to the vacancy will be carried out.
job_city	object	City where the activities related to the vacancy will be carried out.
contract_type	object	Type of contract specified by the employer.
telecommuting	object	Teleworking option indicated by the employer.
disability	object	Option indicating if the vacancy is open to individuals with disabilities.
job_url	object	URL for applying to the vacancy, published by the authorized provider.
hired	object	Confirmation of hiring status, provided by the employer.

Source: Authors' own creation.

Methodology

This proposal employs a combination of clustering, classification, and collaborative filtering algorithms to identify the ten best available vacancies for a given resume. These results are determined by assessing the degree of similarity between the resume and all available vacancies. A detailed description of the methodology is provided in Figure 2. The input data consists of applications submitted from resumes to job vacancies, with the "*hired*" attribute reported by the employer. Only records of applications that resulted in successful recruitment are included.

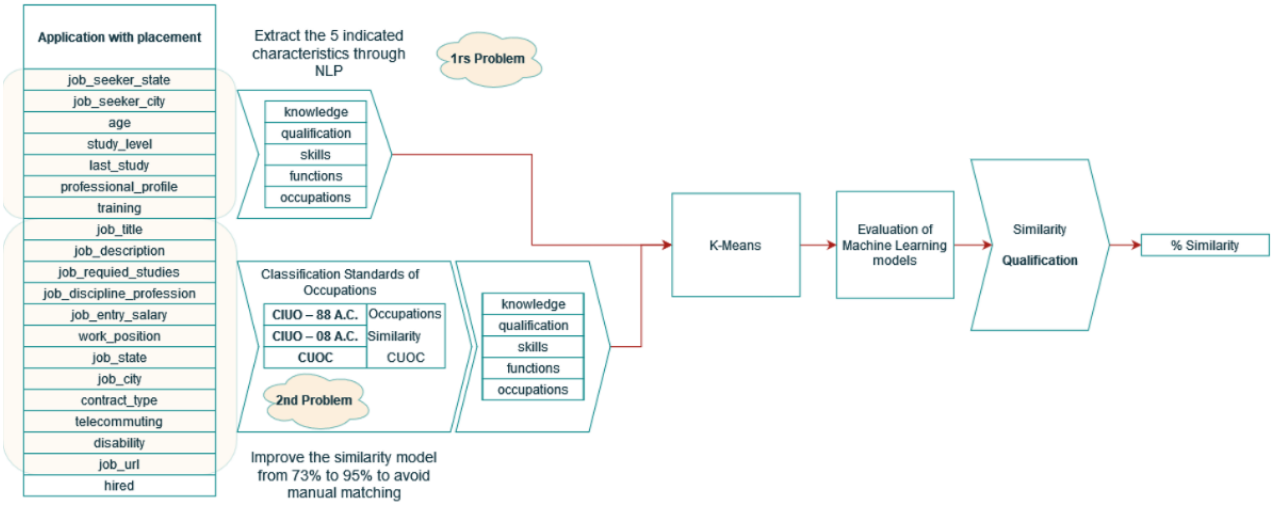

FIGURE 2.
Job system recommendation methodology using the CUOC
Source: Authors' own creation.

To preprocess the job application dataset, it was necessary to divide the dataset into 16 parts according to its size. Each part was preprocessed by multiprocessing pools, using 80 % of the computer's CPUs. This approach facilitated parallel preprocessing through batch files, resulting in faster generation of the final dataset. Once the training data were prepared, the text attributes of the job title and work position were standardized via natural language processing on the basis of the CUOC framework. The following steps were performed during the standardization process:

- Textual data cleaning is the foundational step in text analysis. This process involves removing all extraneous elements, such as punctuation marks, HTML tags, and special characters, to prepare the text for further processing.
- The next step involves processing documents into lexical components. This phase includes segmenting the text into individual words, eliminating unnecessary whitespace, and filtering out stop words to streamline the data.
- Normalization standardizes the text by converting words to their root form and removing accents, ensuring consistency across the dataset.
- Finally, similarity by distance is used when standardizing text with dictionaries. Various word distance methods are applied to measure the similarity between words, aiding in the refinement and accuracy of the text analysis.

To measure similarity by distance, we tested two methods: the Levenshtein distance and Word2Vec [33]. The Levenshtein distance is an algorithm that calculates the distance between two words by determining the minimum number of operations needed to transform one word into another. In contrast, Word2Vec uses neural networks to generate word vectors, enabling the calculation of semantic closeness. While the results obtained with Word2Vec were very similar to those from the Levenshtein distance, it required more processing time. Consequently, the Levenshtein distance was used for this study. Several libraries capable of calculating this distance were evaluated, and the one with the shortest processing time was chosen. This method was applied during the preprocessing of vacancies and in the standardization of text from dictionaries. At this stage, the matching threshold was set to accept similarities with a minimum of 96 %, an adjustment that performed best after several iterations.

During processing, to obtain the initial similarity rating, the Levenshtein distance algorithm was used without hyperparameter adjustment and without training, as the processing was conducted in real time. The

distances obtained through the ratio calculation were used for the initial sorting of the top 50,000 similarity results.

When the standardization process is complete, the attributes of knowledge, qualifications, skills, functions, and occupations that must be fulfilled to perform the position are extracted from the occupation according to the CUOC framework. Note that these attributes are not explicitly stated in the resume, and therefore improve the input data.

In the next step, the information from the applications was grouped to obtain clusters of similarities between resumes and vacancies. This research uses the K-means clustering algorithm, which is chosen for its computational efficiency compared with other clustering algorithms, and its effectiveness in grouping large datasets [2]. The resulting clusters were used as the independent variables for training the machine learning models that were evaluated. After obtaining the groups from the clustering algorithm, we evaluated three machine learning models, namely, KNN, DNN, and AdaBoost. The best model results in the class in which the resume best relates to the group of vacancies. This generates a reduction in the dataset that will be evaluated in the recommendation system.

The recommendation system was implemented via two similarity models. The first model is based on the Levenshtein distance, which assigns greater weights to the attributes listed in Table 3, represented by the "*levenshtein*" attribute. The second model employs user-based collaborative filtering via cosine similarity, which is represented by the "cosine" attribute. Consequently, a sample of the top 50,000 records was extracted on the basis of the "*levenshtein*" value. The second model was subsequently evaluated, and the top 10 results were identified by ranking them according to both "*cosine*" and "*levenshtein*".

TABLE 3.
Attributes with major weights

Job offer	Resume
knowledge	knowledge
qualification	qualification
skills	skills
functions	functions
occupations	occupations
job_state	job_seeker_state
job_city	job_seeker_city
work_position	last_study
	professional_profile

Source: Authors' own creation.

On the basis of the data available for this study, and to achieve the best results, we selected collaborative filtering recommendation systems. This decision was made because content-based and hybrid models require the job seeker's preference history, which is not publicly accessible, and is maintained by individual employment platforms.

Results

Experiments

As a first step, the data with CUOC attributes are grouped via the K-means algorithm, with the number of groups determined via the elbow curve method and the silhouette plot. The results are shown in Figure 3.

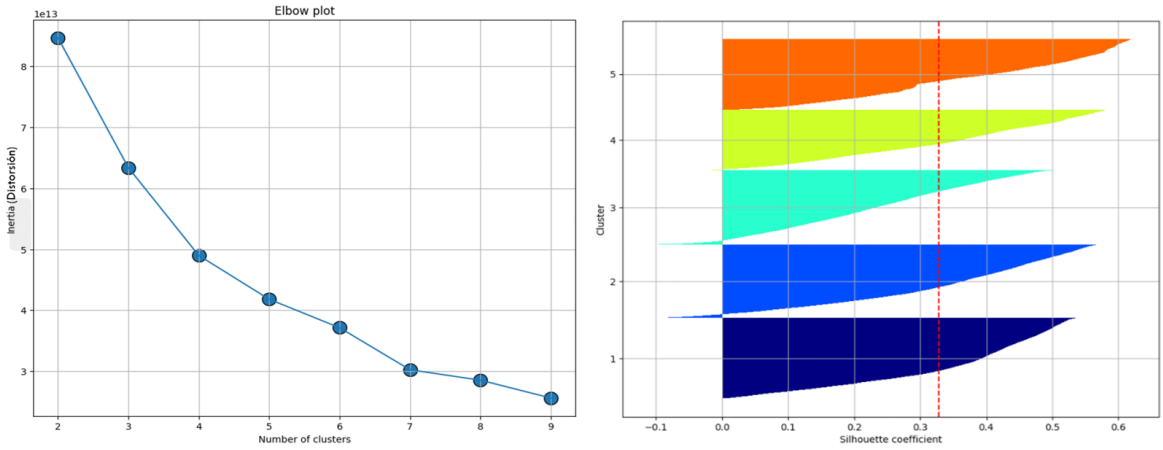

FIGURE 3.
Graphical validation of the K-means hyperparameters via the elbow curve and the silhouette graph
Source: Authors' own creation.

Afterward, the remaining hyperparameters were iteratively adjusted, and after several combinations, the best silhouette coefficient of 0.3078 was achieved with the hyperparameters detailed in Table 4.

TABLE 4.
K-means hyperparameters

Hyperparameter	Value
n_clusters	5
init	K-means++
n_init	10
max_iter	250
random_state	0

Source: Authors' own creation.

The resulting groups were used as the Y-label, or dependent variable, for the evaluation of the selected classification models. Then, we conducted experiments without using the CUOC framework, applying a cross-validation approach during the training process. The results are shown in the learning curves of the validation data presented in Figure 4.

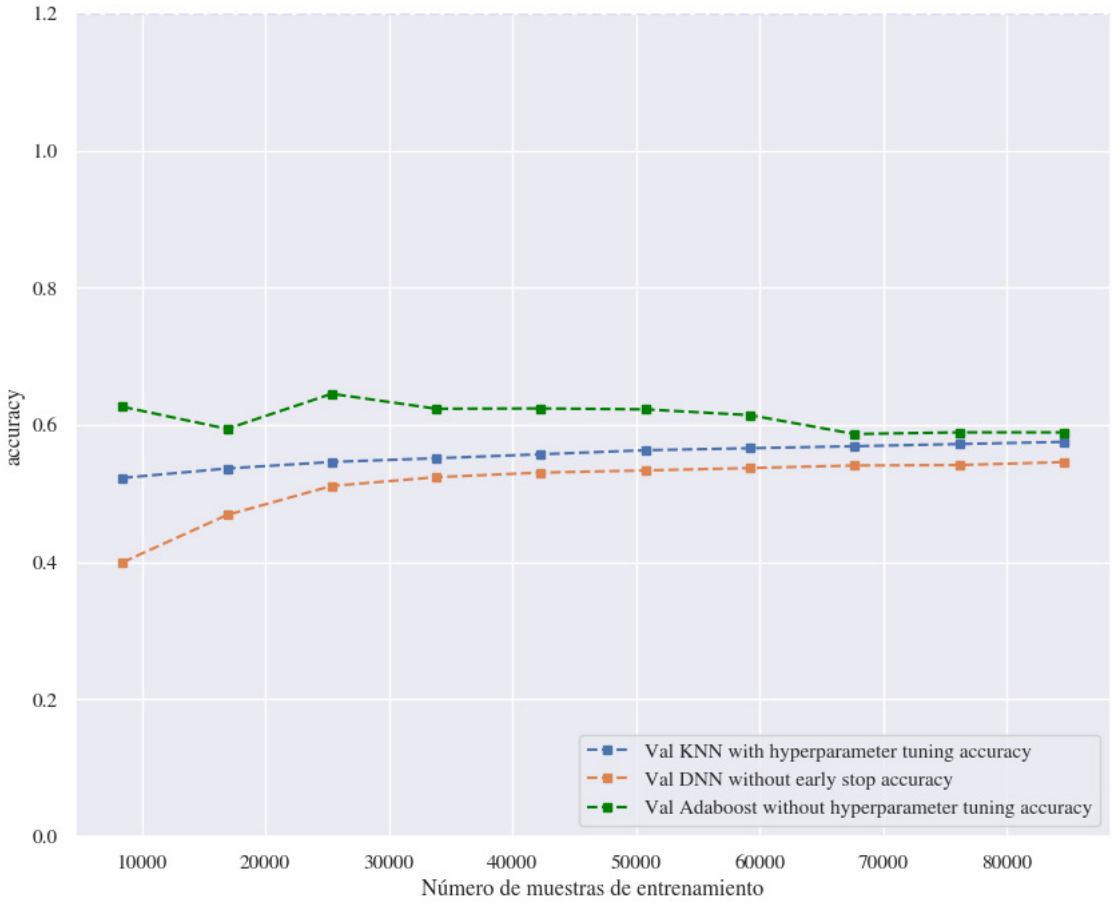

FIGURE 4.
Learning curve of the validation data without CUOC
Source: Authors' own creation.

Figure 5 shows the learning curves from the validation data for each evaluated model via the CUOC framework with hyperparameter tuning.

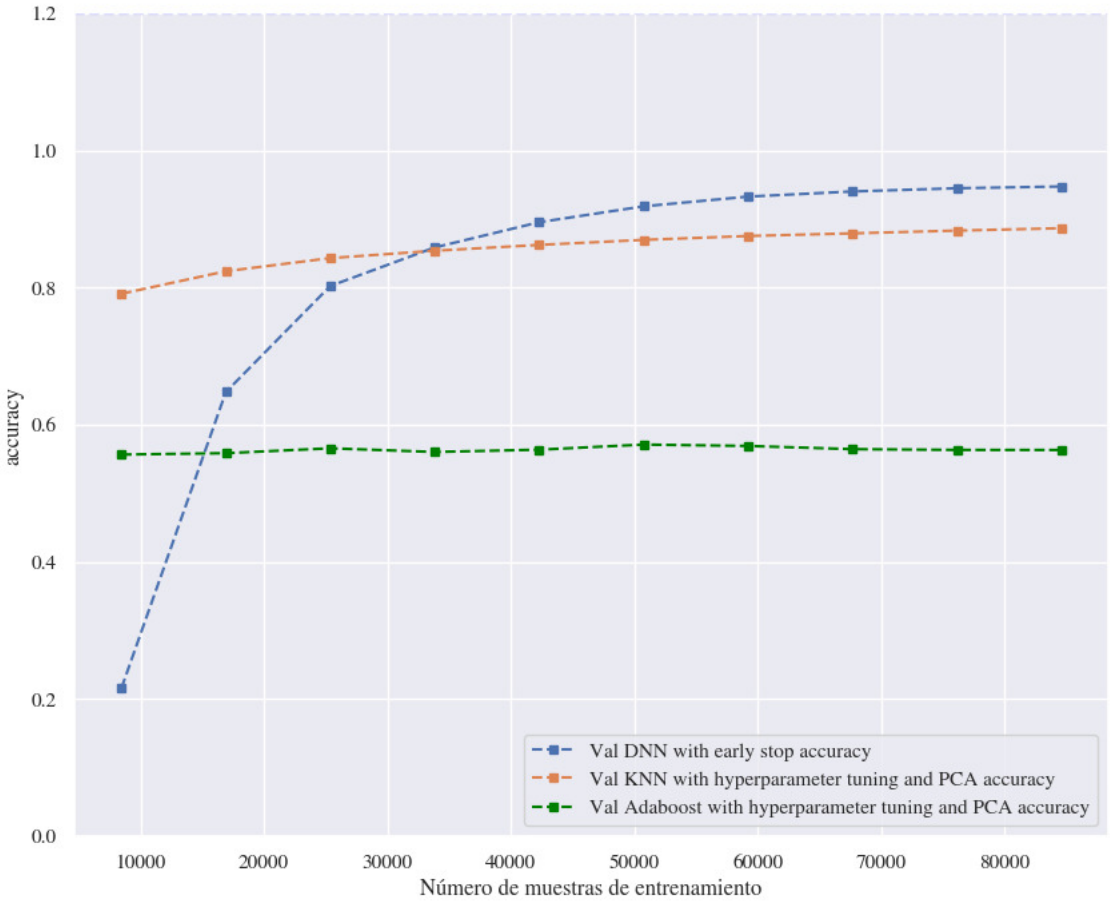

FIGURE 5.
Learning curves of the validation data via CUOC
Source: Authors' own creation.

Hyperparameter tuning was performed via the *GridSearchCV* method, which finds the best hyperparameters through an exhaustive search. Similarly, the models were tested without hyperparameter adjustment to evaluate whether they yielded better results with the suggested values. The best results are presented in Table 5.

TABLE 5.
Best results with CUOC

Algorithm	Accuracy	Precision	Recall	F1-score	Time (sec)	Overall
DNN without early stop	0.99	0.99	0.99	0.99	2426.61	0.99
DNN with early stop	0.99	0.99	0.99	0.99	2495.16	0.98
AdaBoost without hyperparameter tuning	0.94	0.94	0.94	0.94	0.13	0.94
XAdaBoost with hyperparameter tuning and PCA	0.90	0.89	0.89	0.89	1.76	0.89
KNN without hyperparameter tuning	0.87	0.86	0.86	0.86	1.61	0.86
KNN with hyperparameter tuning	0.86	0.84	0.84	0.84	1.64	0.84
KNN without hyperparameter tuning and PCA	0.83	0.84	0.84	0.84	1.92	0.84
KNN with hyperparameter tuning and PCA	0.80	0.79	0.73	0.73	0.13	0.76

Source: Authors' own creation.

On the basis of these results, AdaBoost without hyperparameter tuning was selected as the best model. This decision was made because it provides strong accuracy within a short execution time, making it ideal for prompt recommendations. Importantly, in this case, accuracy is the most relevant metric, as it reflects the number of correctly predicted cases, both positive and negative. Overall, the combination of all the metrics determines the most effective model. The specific improvements in these results compared with those of previous studies are presented below.

- Use of multiprocessing for parallel batch processing. This allows for faster result generation and enables the processing of more information in real time compared to the current platforms.
- Compared with previous methods, the use of the CUOC framework contributes to improved precision, recall, and F1 scores. For the best model without CUOC, the DNN achieved an accuracy of 0.63, a recall of 0.46, and an F1 score of 0.53. With CUOC standardization, the best model

achieved an accuracy of 0.94, a recall of 0.94, and an F1 score of 0.94, demonstrating superior performance in terms of job recommendations in the Colombian context. Figure 6 presents a comparison of the learning curves for the best model with CUOC and the same model without CUOC.

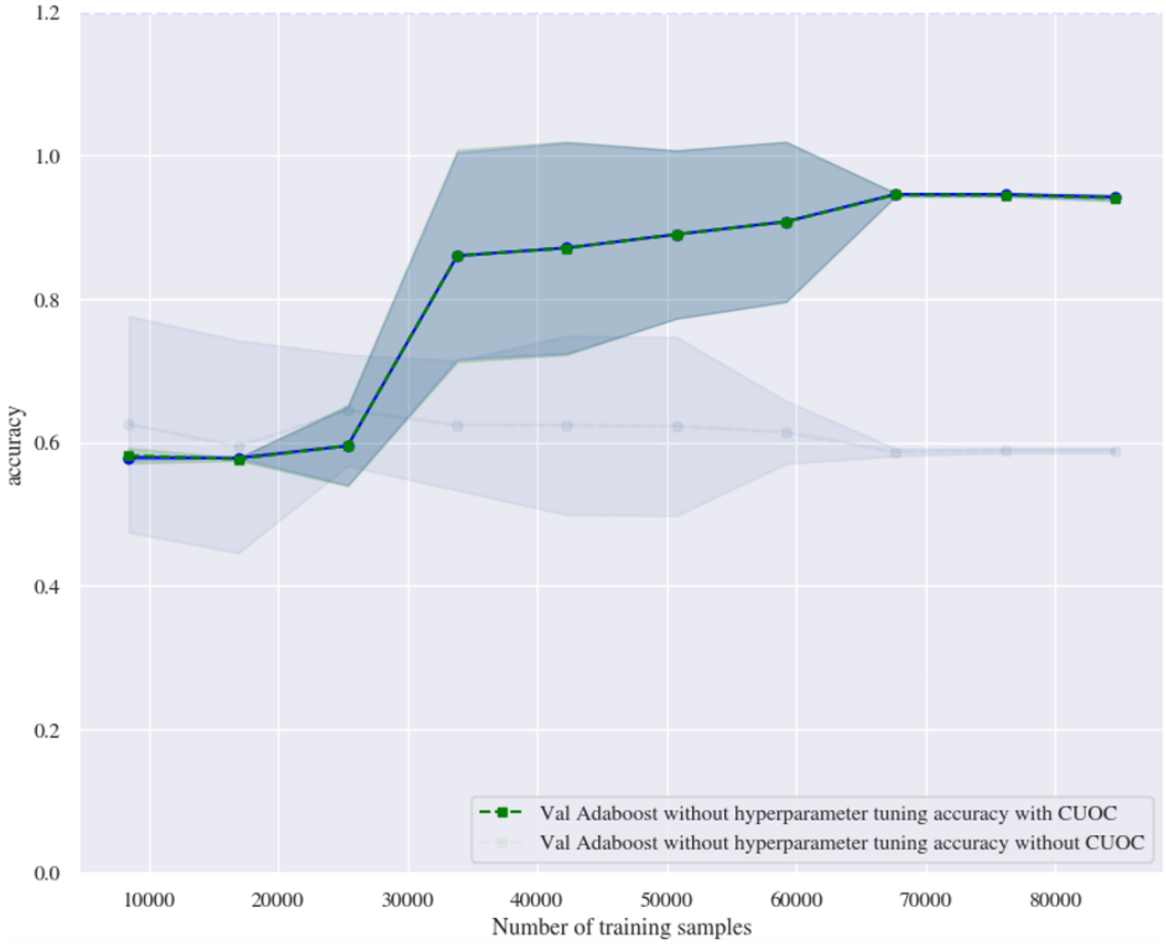

FIGURE 6.

AdaBoost without hyperparameter tuning learning curves compared using CUOC and not using CUOC

Source: Authors' own creation.

Web application architecture

The proposed web application was developed via the Django framework, which allows Python to be used as the base language. It uses a Model-View-Template architecture. The architecture includes the *index.html* template with fields for state, city, and study level; an autofill field for last study; a text box for the job profile; and a search vacancy button, as shown in Figure 7.

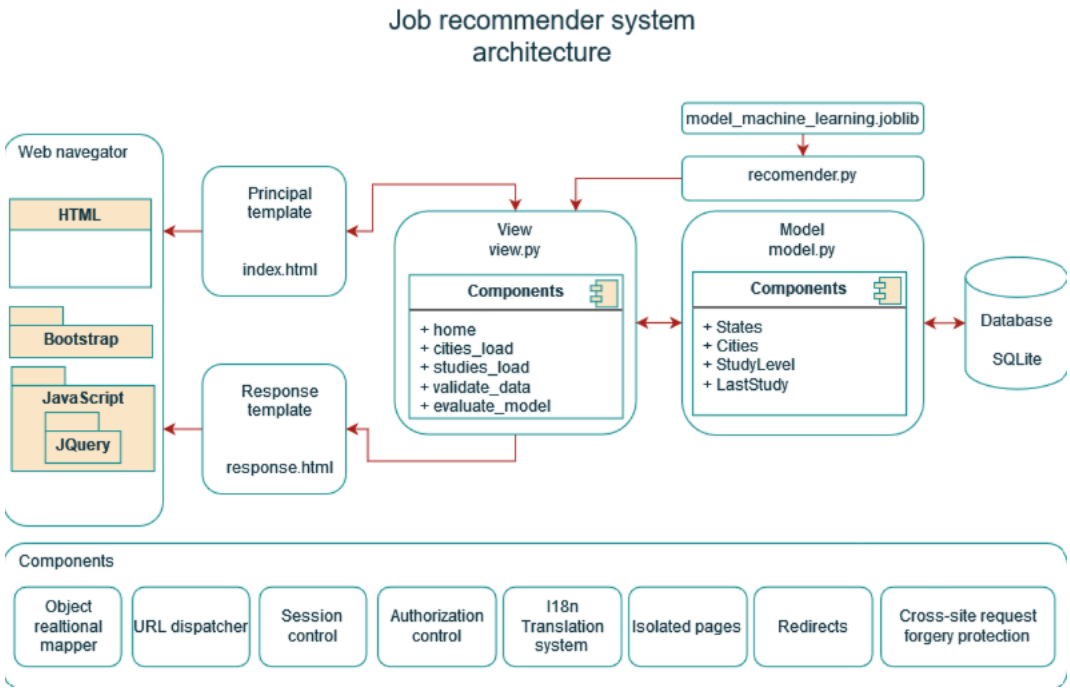

FIGURE 7.
Job recommender system architecture
Source: Authors' own creation.

When the main page is loaded, the *index.html* template is called. It uses markup language to display the information sent by the view through the home (request) function. This function involves the model components: states, cities, study level, and last study. Once the information is fully populated, a request to calculate the similarities is sent to the view. The view then calls the *Recommendator.py* model class, which executes the process described in Figure 2 and returns the top 10 results via the *response.html* template.

Validation

To validate the recommender system, data from professional networks and job portals were used. Vacancy details were sourced from the SPE's single employment portal, whereas data for validation were retrieved from LinkedIn via its API. This information is manually entered into the user interface and compared against the recommendations generated by the professional network. Figure 8 illustrates the process of data capture and collection. The results of the top 10 recommendations are returned to the job seeker in a list of vacancies starting with the greatest similarity.

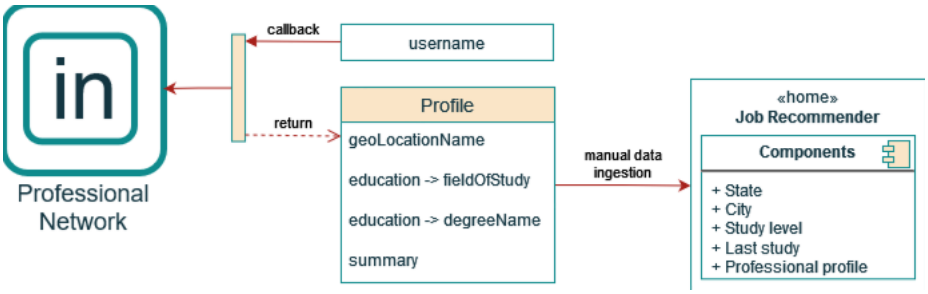

FIGURE 8.
Consuming attributes for validation via the LinkedIn API
Source: Authors' own creation.

The top 10 recommendations are presented to the job seeker as a list of vacancies, ranked by highest similarity. This is achieved via a collaborative filtering recommendation model, which calculates user ratings via neighborhood-based methods, specifically the user-based methods [3]. Item-by-item similarity is then applied to generate recommendations from a rating matrix. To enhance the results from the collaborative filtering-based recommendation, the Levenshtein distance similarity is included. This method calculates the similarity between two strings, addressing variations in writing. A similarity threshold of 90 % is applied, resulting in a classification variable called "*similarity*".

Finally, the information is entered into the recommendation model, where a vector based on word frequency is used, giving greater weight to meaningful words. Stop words in the Spanish language were removed for this step. The cosine similarity between the vectors is then measured, creating a similarity matrix. The recommendations are calculated from the similarity matrix, which produces a score for the resume. The score is stored, and the top 10 vacancies with the highest similarity and score are displayed, as shown in Figure 9.

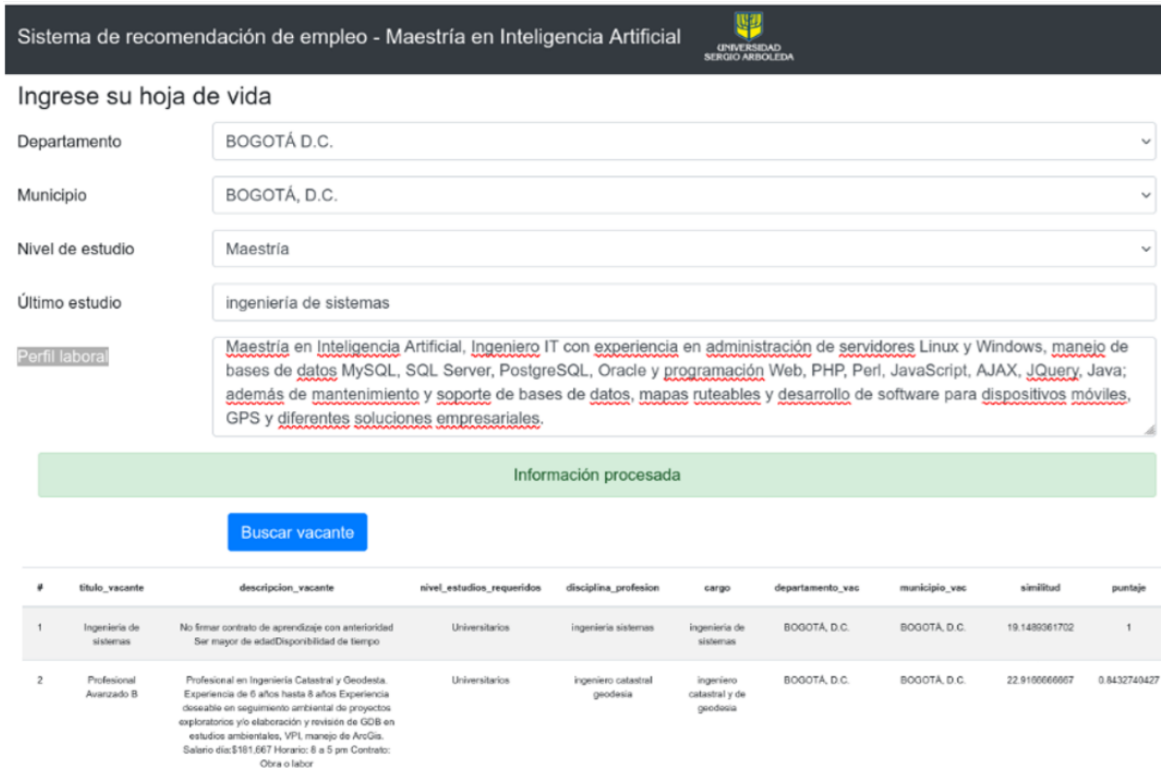

FIGURE 9.
Visualization of the results from the employment recommendation system
Source: Authors' own creation.

Note that the privacy of the jobseekers' data is not at risk, as the proposed system does not request private or sensitive information. It does not ask for data such as gender, race, age, or residence address. The platform allows users to select their desired geographical location for work, and only requires information on the level of study, last completed study, and a description of their work profile. At the end of the search process, the files containing the data entered through the main form are automatically deleted.

The proposed system has a limitation, in that the available vacancies must be preprocessed in advance for standardization to avoid hindering the resumption of the recommendation process. This is due to the large number of vacancies that may be available per candidate. Given the dynamism of the Colombian labor market, the number of available vacancies for application may range from 150,000 to approximately 350,000 nationwide. The current system architecture allows for the evaluation of a single resume per session, as the expected result is a list of the 10 best vacancies for the entered profile.

The back end of the proposed system requires at least Python version 3.11, SQLite version 3, PyPI sources, a separate virtual environment configuration, at least 16 GB of RAM, and 20 processing cores. The front end of the system can be ported if it can be connected to a web service on the server where the preprocessed vacancies are maintained. When deployed on the web through Django, it can be accessed by any system operating with a web browser.

Conclusions

The research presented in this study shows that including CUOC characteristics in both resumes and job vacancies increases the likelihood that job seekers will find a suitable position that aligns with their profile and achieve a better employment relationship within the context of the Colombian labor market.

During the study, a problem was identified with data standardization and cleanliness. The best way to achieve data homogeneity was to use natural language processing. This approach facilitated the inclusion of implicit attributes within the occupation categories outlined in the CUOC. The recommendation system successfully preserved diversity in job recommendations, thereby mitigating potential biases through stratification in the training and testing datasets.

The model utilizing the CUOC framework achieved strong performance in terms of precision, recall, F1 score, and overall evaluation metrics, with all the metrics showing a value of 0.94. Compared with the baseline model without CUOC, the new recommendation system demonstrated a 15 % improvement in accuracy, highlighting the effectiveness of the applied machine learning algorithms.

In the Colombian context, this approach holds significant potential for application across various sectors. In the education sector, academic profiles can be assessed against university program offerings to guide students toward the most suitable career paths, fostering success both academically and professionally. Similarly, in the employment sector, this method can enhance the existing government job portal by analyzing job seekers' preloaded resumes and matching them to the most appropriate government vacancies. Finally, in the sports sector, this framework could be adapted by adjusting the evaluation criteria to assess aspiring athletes, helping identify the sport best suited to their physical capabilities and soft skills.

When considering the computational cost of using large NLP models such as BERT versus simpler similarity functions in contexts where the analyzed text is short, such as in this study, it is crucial to weigh the benefits against the financial implications of their implementation in production. Although large pretrained models such as BERT provide sophisticated semantic understanding, and can capture nuanced language patterns that improve the quality and relevance of matching resumes with job descriptions, they incur significant computational expenses. These models require substantial processing power and memory, especially during inference, which leads to higher costs for cloud services and hardware maintenance in production. Conversely, simpler similarity functions, while less sophisticated in capturing deep contextual relationships, often produce results of comparable quality when handling shorter texts.

References

[1] Technical Annex of the Unique Classification of Occupations for Colombia (CUOC), National Administrative Department of Statistics of Colombia (DANE), Aug 2022. [Online]. Available: https://www.dane.gov.co/files/sen/nomenclatura/cuoc/documento-clasificacion-unica-ocupaciones-colombia-CUOC-2022.pdf.

[2] M. H. H. Hisham, M. A. A. Aziz, and A. A. Sulaiman, "Job classification: A Review of Data, Features, and Methods", Nov 2022.

[3] C. C. Aggarwal and Others. Recommender systems, volume 1. Springer, 2016.

[4] S. T. Al-Otaibi. A survey of job recommender systems. International Journal of the Physical Sciences, 7:5127–5142, 7 2012.

[5] E. Yıldırım, P. Azad, and Şule Gündüz Öğüdücü. bideepfm: A multi-objective deep factorization machine for reciprocal recommendation. Engineering Science and Technology, an International Journal, 24:1467–1477, 12 2021.

[6] C. de Colombia. Ley 1636 de 2013 - mecanismo de protección al cesante en Colombia. 6 2021.

[7] T. Schmitt, F. Gonard, P. Caillou, and M. Sebag. Language modelling for collaborative filtering: Application to job applicant matching. pages 1226–1233. IEEE, 11 2017.

[8] R. Boselli, M. Cesarini, F. Mercorio, and M. Mezzanzanica. Classifying online job advertisements through machine learning. Future Generation Computer Systems, 86:319–328, 9 2018.

[9] E. Colombo, F. Mercorio, and M. Mezzanzanica. Ai meets labor market: Exploring the link between automation and skills. Information Economics and Policy, 47:27–37, 6 2019.

[10] E. Lacic, M. Reiter-Haas, D. Kowald, M. R. Dareddy, J. Cho, and E. Lex. Using autoencoders for session-based job recommendations. User Modeling and User-Adapted Interaction, 30:617–658, 9 2020.

[11] S. U. Habiba, M. K. Islam, and F. Tasnim. A comparative study on fake job post prediction using different data mining techniques. pages 543–546. IEEE, 1 2021.

[12] T. K, U. V, S. M. Kadiwal, and S. Revanna. Design and development of machine learning based resume ranking system. Global Transitions Proceedings, 3:371–375,2022.

[13] A. Talun, P. Drozda, L. Bukowski, and R. Scherer. FastText and XGBoost Content-Based Classification for Employment Web Scraping. Springer International Publishing, 2020.

[14] S. Pudasaini, S. Shakya, S. Lamichhane, S. Adhikari, A. Tamang, and S. Adhikari. Application of NLP for Information Extraction from Unstructured Documents. Springer Singapore, 2022.

[15] B. Parida, P. Kumar Patra, and S. Mohanty. Prediction of recommendations for employment utilizing machine learning procedures and geo-area based recommender framework. Sustainable Operations and Computers, 3:83–92, 2022.

[16] Z. Tasnim, F. M. J. M. Shamrat, S. M. Allayear, K. Ahmed, and N. I. Nobel. Implementation of an Intelligent Online Job Portal Using Machine Learning Algorithms. Springer Singapore, 2021.

[17] S. Okura, Y. Tagami, S. Ono, and A. Tajima. Embedding-based news recommendation for millions of users. pages 1933–1942. ACM, 8 2017.

[18] C. Qin, H. Zhu, T. Xu, C. Zhu, C. Ma, E. Chen, and H. Xiong. An enhanced neural network approach to person-job fit in talent recruitment. ACM Transactions on Information Systems, 38:1–33, 3 2020.

[19] S. Jia, X. Liu, P. Zhao, C. Liu, L. Sun, and T. Peng. Representation of job-skill in artificial intelligence with knowledge graph analysis. pages 1–6. IEEE, 12 2018.

[20] R. Mishra and S. Rathi. Enhanced dssm (deep semantic structure modelling) technique for job recommendation. Journal of King Saud University - Computer and Information Sciences, 34:7790–7802, 10 2022.

[21] S. Nasser, C. Sreejith, and M. Irshad. Convolutional neural network with word embedding based approach for resume classification. pages 1–6. IEEE, 7 2018.

[22] J. Jiang, S. Ye, W. Wang, J. Xu, and X. Luo. Learning effective representations for person-job fit by feature fusion. pages 2549–2556. ACM, 10 2020.

[23] Y. Luo, H. Zhang, Y. Wen, and X. Zhang. Resumegan. pages 1101–1110. ACM, 11 2019.

[24] A. M. Elkahky, Y. Song, and X. He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. pages 278–288. International World Wide Web Conferences Steering Committee, 2015.

[25] S. Benabderrahmane, N. Mellouli, M. Lamolle, and N. Mimouni. When deep neural networks meet job offers recommendation. pages 223–230. IEEE, 11 2017.

[26] Y. Deng, H. Lei, X. Li, and Y. Lin. An improved deep neural network model for job matching. pages 106–112. IEEE, 5 2018.

[27] S. Zhang, L. Yao, A. Sun, and Y. Tay. Deep learning-based recommender system: A survey and new perspectives. ACM Computing Surveys, 52:1–38, 1 2020.

[28] T. V. Huynh, K. V. Nguyen, N. L.-T. Nguyen, and A. G.-T. Nguyen. Job prediction: From deep neural network models to applications. pages 1–6. IEEE, 10 2020.

[29] S. A. Chala, F. Ansari, M. Fathi, and K. Tijdens. Semantic matching of job seeker to vacancy: a bidirectional approach. International Journal of Manpower, 39:1047–1063, 11 2018.

[30] L. Duan, X. Gui, M. Wei, and Y. Wu. A resume recommendation algorithm based on k-means++ and part-of-speech tf-idf. pages 1–5. ACM Press, 2019.

[31] D. Mhamdi, R. Moulouki, M. E. Ghoumari, M. Azzouazi, and L. Moussaid. Job recommendation based on job profile clustering and job seeker behavior. Procedia Computer Science, 175:695–699, 2020.

[32] W. Chen, X. Zhang, H. Wang, and H. Xu. Hybrid deep collaborative filtering for job recommendation. pages 275–280. IEEE, 9 2017.

[33] Church, Kenneth Ward. Word2Vec. Natural Language Engineering, 2017, vol. 23, no 1, p. 155-162.

Notes

* Research article.

How to cite this article: C. M. Caro Cortés, J. P. Ospina López, "A Job Recommender System for the Unique Framework of Classification of Occupations in Colombia (CUOC) Via Collaborative Filtering" Ing. Univ. vol. 28, 2024. https://doi.org/10.11144/Javeriana.iued28.jrsu