Received 16 August 2024, accepted 19 October 2024, date of publication 23 October 2024, date of current version 1 November 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3485111*

RESEARCH ARTICLE

# A Seamless ChatGPT Knowledge Plug-in for the Labour Market

**RUBÉN ALONSO[1,2], DANILO DESSÍ[3], ANTONELLO MELONI[4], MARCO MURGIA[4], DIEGO REFORGIATO RECUPERO[4], AND GIUSEPPE SCARPI[1]**

[1]R2M Solution s.r.l., 27100 Pavia, Italy
[2]Programa de Doctorado, Centro de Automática y Robótica, Universidad Politécnica de Madrid-CSIC, 28500 Madrid, Spain
[3]Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah, United Arab Emirates
[4]Department of Mathematics and Computer Science, University of Cagliari, 09124 Cagliari, Italy

Corresponding author: Diego Reforgiato Recupero (diego.reforgiato@unica.it)

This work was supported in part by the H2020 project Safe and Trusted Human Centric Artificial Intelligence in Future Manufacturing Lines (STAR) for dynamic and unpredictable manufacturing environments under Grant 956573; in part by European Union's Horizon 2020 Research and Innovation Programme under Grant 101058174 (TRINEFLEX); in part by EU's LIFE Programme under Grant 101138427 (MULTIMOLD); and in part by European Union's Horizon Europe Research and Innovation Programme under Grant 101069364 (NGI-SEARCH, framed under Next Generation Internet Initiative).

**ABSTRACT** In today's rapidly evolving labor market, the emergence of new roles and the decline of traditional ones have led to a complex landscape of job titles and skill requirements. This complexity often causes ambiguity and confusion, affecting both novices and experienced professionals. To address this, extensive international efforts have produced reference databases of jobs and skills, such as ESCO and O*NET. However, the challenge remains to make this information easily accessible and interpretable for users with varying levels of expertise. To address the challenge above, this paper introduces a Knowledge Plug-in for ChatGPT, designed to serve as an intuitive, user-friendly interface between workers and these authoritative databases. By harnessing the power of natural language processing (NLP), the plug-in enables a seamless question-answering experience, effectively masking the underlying complexity with a carefully engineered architecture. Furthermore, generative AI enhances the user experience by providing relevant information in domains extending beyond the traditional scope of employment. An initial user study demonstrates the plug-in's effectiveness in improving the usability and accuracy of job-related queries. We detail the development, architecture, and validation of this innovative tool, highlighting its potential impact on the future of employment search and career development.

**INDEX TERMS** Conversational agents, labor market, large language models, natural language processing, occupational databases.

## I. INTRODUCTION

The job market, also referred to as the labor market, serves as the dynamic arena where employers seek suitable candidates, and individuals pursue employment opportunities. This ecosystem is profoundly influenced by the intricate interplay between the demand for labor and the availability of workers within the broader economy.

According to a recent survey conducted by the Bureau of Labor Statistics (BLS), substantial hiring has been observed since 2021, with a projected growth of 8.3 million jobs from 2021 to 2031.[1]

The evolving nature of society continually reshapes the demand for various professions. Job seekers are advised to tailor their skills to align with the growing sectors, ensuring they secure well-paying positions in industries with substantial growth and advancement potential.

Numerous online systems facilitate the process of uploading or creating resumes and automatically searching for jobs tailored to individual qualifications. Prominent examples of

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Da Lin.

[1]Source: Bureau of Labor Statistics https://www.bls.gov/emp/

such platforms include Monster.com,[2] OfferZen,[3] LinkedIn,[4] CareerBuilder,[5] Glassdoor,[6] JobStreet,[7] and ZipRecruiter,[8] each boasting millions of job listings and resumes from potential candidates.

While these platforms simplify the job search process, securing the most promising opportunities is far from straightforward. For instance, a candidate possessing computer science skills may not be aware that a position as energy community CTO also requires expertise in fundraising, proposal writing, and startup development. As a consequence, they could be rejected as their resume lacks crucial elements.

Addressing such nuanced requirements is challenging, and obtaining precise information on how to acquire the necessary skills can be elusive. Providing accurate responses to questions like these necessitates relying on data from reputable organizations capable of consistently updating information. Additionally, job-related data must be mapped to values and normalized to enable querying with a certain degree of granularity.

The primary challenge lies in efficiently and consistently sourcing accurate information. A recent avenue that has shown promise involves leveraging conversational agents, which have witnessed significant technological advancements [1] due to the integration with Large Language Models (LLMs). Notably, the emergence of ChatGPT,[9] based on a Generative Pre-trained Transformer as LLM, has demonstrated impressive conversational capabilities across various domains. However, despite claims that ChatGPT can answer follow-up questions, acknowledge mistakes, challenge incorrect premises, and reject inappropriate requests, it occasionally generates responses that may be misleading or inaccurate. OpenAI, the organization behind ChatGPT, has acknowledged certain limitations: i) occasional generation of plausible-sounding but nonsensical or incorrect answers; ii) sensitivity to tweaks in input phrasing or repeated prompts; iii) verbosity and overuse of certain phrases; iv) failure to seek clarification for ambiguous user queries; v) potential biased behavior in response to inappropriate requests. For instance, if a user queries ChatGPT multiple times about the top $n$ essential skills for a specific job, it consistently produces varying responses, with skills presented in different orders or entirely different skill sets.

The reliability of ChatGPT has been further questioned by authors in [2], particularly within the scholarly domain. The study highlighted similar issues, noting instances where ChatGPT partially invented elements of responses, leaving users without clear guidance on correctness. Consequently, while ChatGPT may offer correct responses at times, it cannot be wholly relied upon, necessitating the exploration of alternative mechanisms. Similar challenges persist in other LLMs such as LLaMa 3,[10] Koala,[11] Alpaca,[12] Mixtral,[13] and Phi-3.[14] As the pursuit of reliable information continues, it remains crucial to critically evaluate and complement the outputs of these advanced language models with additional verification mechanisms.

To address the aforementioned challenges, requirements, and constraints associated with ChatGPT, and inspired by the objectives outlined in [2], this paper introduces an innovative solution: a seamlessly integrated knowledge plug-in designed specifically for ChatGPT. The primary aim of this plug-in is to enhance ChatGPT's capabilities, enabling it to effectively respond to targeted queries within the job market domain. This proposed enhancement seeks to overcome the identified limitations and significantly augment the overall utility and performance of ChatGPT in addressing specialized questions related to employment and career topics.

We have harnessed a robust knowledge base comprising data from two authoritative sources: O*NET[15] and ESCO.[16] The O*NET database offers comprehensive information on occupations, skills, and work-related attributes, while ESCO, the European Classification of Skills, Competences, Qualifications, and Occupations, provides a standardized framework for understanding and classifying skills and qualifications within the European context.

By leveraging the wealth of data from these reputable sources, our knowledge base is enriched with up-to-date and diverse information on the job market, ensuring a robust foundation for the proposed seamless knowledge plug-in. This strategic integration not only enhances the accuracy and relevance of ChatGPT's responses but also ensures alignment with both national and European perspectives on the labor market, offering users a comprehensive and globally informed experience.

In particular, our tool is designed to effectively address actions and respond to questions related to the labor domain, leveraging the extensive knowledge sourced from O*NET and ESCO, such as:

- *List some alternative names for the job of Computer Scientist*;
- *List jobs related to Software Engineering*;
- *List the top five skills required by a Data Engineer*;
- *What is the description of a Mechanical Engineer?*
- *Find the similarities between a Mechanical Engineer and a Civil Engineer*;

[2] https://www.monster.com/
[3] https://www.offerzen.com
[4] https://www.linkedin.com/
[5] https://www.careerbuilder.com/
[6] https://www.glassdoor.com/
[7] https://www.jobstreet.com/
[8] https://www.ziprecruiter.co.uk
[9] https://openai.com/blog/chatgpt/
[10] https://llama.meta.com/
[11] https://bair.berkeley.edu/blog/2023/04/03/koala/
[12] https://crfm.stanford.edu/2023/03/13/alpaca.html
[13] https://huggingface.co/docs/transformers/en/model_doc/mixtral
[14] https://huggingface.co/collections/microsoft/phi-3-6626e15e9585a200d2d761e3
[15] https://www.onetcenter.org/database.html#individual-files
[16] https://joinup.ec.europa.eu/collection/labour-market-interoperability/about-esco-european-classification-skills-competences-qualifications-and-occupations-esco

- *List the jobs that require Microsoft Office as a technical skill*;
- *Recommend me some materials about social perceptiveness*.

The responses provided are deterministic and align seamlessly with the information stored in the designated knowledge bases. To delve into greater detail, this paper introduces a set of noteworthy contributions.

- We introduce an architecture for the proposed knowledge plug-in, designed to smoothly integrate with ChatGPT.
- We present an innovative approach to identify if the user is asking something related to the labor domain and the arguments from his/her question in natural language needed to compose a SQL query to the database containing the knowledge. This is achieved through the utilization of the function calling feature, as detailed in the OpenAI documentation.[17] By adopting this methodology, our system deterministically responds to each user query.
- To provide the knowledge, we leverage the taxonomies provided by O*NET and ESCO.
- Our system supports multilingualism, the plug-in will answer in the same language as the question, so not only in English.
- We make publicly available the demo of our system at http://starbot.r2m.cloud[18] as well as the source code and the evaluation results.[19]
- In addition to addressing the technical aspects, our study includes an evaluation of precision across various versions of each proposed query in four distinct languages. Furthermore, we conducted a user study involving 9 participants with expertise in the field, assessing the efficacy of the knowledge plug-in seamlessly integrated into a chatbot.

The remainder of this paper is organized as follows. In Section II, we delve into related works on conversational agents within the labor market domain. Section III provides a comprehensive overview of the O*NET and ESCO databases utilized in this study. Section IV presents the defined architecture and describes each constituent module. Section V illustrates the practical application of our system through an exemplar case. The precision analysis and the user study are conducted and documented in Section VI, which includes information about the annotators and their valuable feedback. Section VII includes a discussion on the proposed knowledge plug-in indicating limitations and possible use cases. Finally, Section VIII concludes the paper by summarizing findings and presenting future directions for our work where we are headed.

[17]https://platform.openai.com/docs/guides/function-calling

[18]User: user1 Password: 7eb115

[19]http://192.167.149.18/A-Seamless-ChatGPT-Knowledge-Plugin-about-the-labor-market-main.zip

## II. RELATED WORK

This paper investigates the integration of artificial intelligence (AI) tools as well as domain-specific knowledge injection into conversational agents for real-world applications in the job market. Recent research has examined how AI agents can help job seekers find employment, upskill, and connect with employers, as well as support employers in filling vacancies using innovative tools [3]. The literature primarily focuses on three commonly used recommendation system approaches: collaborative filtering [4], similarity-based methods [5], and knowledge-based systems [6], often employing a combination of these techniques. For instance, Lu et al. [7] proposed a hybrid recommendation system framework that matches candidates' resumes using content-based similarity while incorporating user interactions through a graph representation. Mishra and Rathi [8] presented a Deep Semantic Structure Modelling (DSSM) approach to represent resumes and job postings, building a semantic embedding space. They used cosine similarity within this space to recommend jobs based on skill and title matches. However, their work overlooks the fact that essential skills for specific jobs are often known beforehand, a piece of information that could enhance the embedding space representation. Similarly, authors in [9] explored joint representation of resumes and jobs using convolutional neural networks (CNNs) while in [10] authors proposed a word-level joint semantic space using recurrent neural networks (RNNs) and hierarchical ability-aware attention mechanisms. Both approaches learned latent features to train models able to predict matching labels.

Current state-of-the-art online hiring platforms like LinkedIn, Monster, and Fiverr rely on ranking candidates based on their skills. This approach, as outlined by LinkedIn researchers in [11], utilizes latent representations to match job requirements with candidate profiles, facilitating efficient recruitment. However, despite these advancements, a crucial limitation exists: users cannot easily explore and understand the complex skill and competency requirements of different jobs. This hinders their ability to proactively develop relevant skills and align themselves with evolving job market demands. Our proposed architecture bridges this gap by offering easy access to structured data through a natural language interface (NLI) powered by advanced NLIDB technologies. Specifically, our knowledge plug-in leverages natural language to enable users to naturally interact with O*NET and ESCO resources. NLIDBs have traditionally utilized various approaches, including rule-based matching [12] and, more recently, deep learning techniques [13], [14], [15]. While our solution also employs advanced transformer-based models to identify relevant information within user queries, we deviate from typical NLIDBs by solely using SQL to retrieve entities O*NET and ESCO. This allows us to perform semantic matching between user-specified concepts and retrieved entities, providing a refined and relevant information selection process. This characteristic is new when looking at existing conversational agents for the labor market [16], [17]. For example, there exist conversational agents such as CASExplorer [16] that help people during the decision process and the exploration of required skills and job career opportunities in choosing the best major. CASExplorer explores users' strengths and personality traits and adopts the IBM Watson[20] suite. In [17], the authors investigated a text-based conversational agent for conducting job interviews for entry-level candidates. The authors showed that efficiency and inclusivity are enhanced during the recruitment process due to the conversational agents. However, the use of domain knowledge while conducting the job interviews was not considered. One more work worth mentioning is [18] where GUApp, a conversational agent to search for jobs in a certain geographical location is released. This conversational agent includes a knowledge plug-in that leverages the *Gazzetta Ufficiale*, the official journal of record of the Italian public administration. GUApp exploits the DBpedia[21] knowledge base as well as ISTAT website[22] (ISTAT is the Italian provider of taxonomies about professions organized by sectors) to match entities (e.g., users' skills, wishes, and ambitions) from the users' input and suggest job openings by solving a recommendation task. A limitation of the GUApp authors' approach is the limited geographical area covered by the tool that might require relevant effort to be applied beyond Italian borders.

With the rise of LLMs, conversational agents that assist with tasks related to the labor market have also started to proliferate. For example, in [19], the authors present an LLM-based tool that customizes and tailors a resume for a specific job offer. Another activity where LLMs support workers and students is in career guidance and discovery by identifying their strengths and aptitudes to define a career path [20]. Conversational agents, particularly those empowered by LLMs, play a crucial role in human resources management and recruitment processes for employers. These agents facilitate the initial screening process, consequently reducing recruitment times [21]. They serve various functions, such as engaging with applicants, addressing inquiries, and screening applications [22]. Nevertheless, it is essential to note that instances have been observed, raising concerns about potential bias in candidate screening processes [23].

Our plug-in exploits recent conversational agents empowered by LLMs to seamlessly connect user questions with relevant information from the O*NET and ESCO databases, key resources for the job market covering the US and European geographical areas. This enables users to easily explore the skills and qualifications needed for their desired job positions, empowering them to make informed career decisions.

[20]https://www.ibm.com/watson
[21]https://www.dbpedia.org/
[22]https://www.istat.it/en/

## III. BACKGROUND

When it comes to increasing the reliability of ChatGPT responses, it is important to rely on public and reputable information. In our particular case, these trusted sources of information are the O*NET and ESCO occupational databases. Both databases have been developed by initiatives backed by government agencies and provide up-to-date information on occupations, skills, and competencies. This information allows both the workforce and companies to obtain answers and insights on the needs of the labor market.

O*NET is the reference occupational information database in the United States. This database is the result of the Occupational Information Network program sponsored by the U.S. Department of Labor, Employment & Training Administration, and developed by the National Center for O*NET Development.[23]

Through this database and other tools, O*NET aims to contribute to the understanding of career and occupational pathways in the United States and to support workforce development. The database is revised and updated periodically through interviews with organizations and employees. The data obtained is refined with the review and knowledge of experts, to publish four versions each year. This regular update not only allows us to understand the changes in the labor market but also to keep up to date in terms of emerging needs or changes in the necessary skills.

The database includes a series of entities associated with each occupation. These entities consist of skills, knowledge, abilities, activities, technologies, or tasks among others. Version 28.1 (November 2023) updated 923 occupations and includes thousands of linkages between occupations and the above-mentioned entities (e.g., skills or technologies). The processing of this information and linkages makes it possible to characterize occupations, calculate distances between them, detect gaps, or establish potential training pathways.

ESCO[24] (European Skills, Competences, and Occupations) is a multilingual database, supporting among others all the official languages of the European Union, and the matching between skills and qualifications. The former is useful to adapt the inputs and outputs to the specific needs of a given region and above all to reach a larger number of potential users. The latter also adds the possibility of standardizing the information related to occupations, skills, and qualifications across different countries and sectors.

ESCO is released less frequently than O*NET, with the last update being a minor version in 2022. However, it encompasses more than 3,000 occupations and over 13,000 skills associated with those occupations. The latest version of ESCO also includes a revision of the mapping between ESCO and ISCO-08.

[23]https://www.onetcenter.org/
[24]https://esco.ec.europa.eu/

ISCO[25] (International Standard Classification of Occupations) organizes and categorizes occupations worldwide. Specifically, ISCO classifies occupations into four groups from the most generic to the most detailed. It starts with Major groups. *Professionals* is one of these groups. This Major group splits into several Sub-major groups. For example, a sub-group of *Professionals* is *Science and Engineering Professionals*. These Sub-major groups are divided into Minor groups, and then into Unit groups, which are the most detailed classification. Following with the example. within *Science and Engineering Professionals* there is a Minor group called *Electrotechnology Engineers* and within this a Unit group called *Telecommunications Engineers*. Each occupation in all four levels is assigned a unique code for traceability and referencing purposes. These hierarchies provide a reference model for developing other classifications of occupations, for example by adapting them to the specific characteristics of a country or region. ESCO includes similar hierarchies and crosswalks with ISCO.

It is important to mention that although these are the best-known and most referenced databases, there are similar initiatives in other countries and regions. For example, in Korea the reference classification is KSCO[26] (Korean Standard Classification of Occupations), a categorization of more than 1200 occupations hierarchized in 5 levels, the same as ISCO, but with a more detailed level called *Detailed occupation*. This classification has been updated regularly since the 1970s, with a revision scheduled for 2024. A further relevant example is NOC[27](National Occupation Classification), the national system for the description of occupations in Canada. NOC maintains a regular update program, with major revisions every 10 years, and is available in French and English. It is also linked to other databases and reference systems in Canada, which allow detailed information to be obtained on almost a thousand occupations, including the skills and abilities required for each occupation.

Finally, it is equally relevant to highlight the efforts to find crosswalks. Crosswalks are essential methods and tools for mapping some of these occupational information databases and systems. These mappings allow interoperability and comparison between skills and occupations in different locations. At the level of occupation definition, most databases have an ISO mapping. In recent years, an effort has been made to seek interoperability between O*NET and ESCO, given the importance of both. In 2022, ESCO published the first report [24] detailing the mapping between O*NET and ESCO. This mapping was developed using a hybrid approach based on AI models relying on Bidirectional Encoder Representations from Transformers (BERT) [25], the results of which were later refined by a group of experts.

[25] https://www.ilo.org/public/english/bureau/stat/isco/isco08/

[26] http://kssc.kostat.go.kr/ksscNew_web/ekssc/common/selectIntroduce.do?part=1&bbsId=isco_s

[27] https://noc.esdc.gc.ca/

## IV. ARCHITECTURE

The architecture of the proposed ChatGPT plug-in is visually depicted in Figure 1, illustrating various blocks, each composed of distinct modules.

The components of the proposed knowledge plug-in comprise the Plug-in Interface, the Plug-in Manager, the Input Analyzer, the Answer Constructor, and the Transformers model block. The Plug-in Interface encompasses the REST API block, serving as an interface accessible to any server for invoking back-end functionalities. This design choice fosters a decoupling effect between the knowledge plug-in and potential platforms seeking to leverage its capabilities.

The Plug-in Manager is responsible for orchestrating the processing of incoming input across various back-end submodules.

The Input Analyzer block features distinct modules to identify and gather information essential for constructing user responses. The initial module within the Input Analyzer is the GPT (Generative Pre-trained Transformer) Function calling module, tasked with analyzing input text and extracting parameters specified in the function list. Subsequently, the Parameter Validator module searches for the extracted parameters in O*NET and ESCO, returning the validated ones.

The validated parameters, along with the triggered function's name, are then transmitted from the GPT Function calling module to the Answer Constructor Block, through the Plug-in Manager.

Within the Answer Constructor Block, the Function Factory module encompasses all the functions necessary to generate responses tailored to specific types of questions within the labor domain, as well as questions outside of it. If the user's request activates a function from the designated list, the module delivers a response employing the corresponding domain-specific modules. These include the O-NET database, the ESCO database, and the course recommendation modules.

The first two modules gather data required for inquiries related to jobs, skills, tools, tasks, and work activities. Meanwhile, the course recommendation module provides suggestions for courses that may enhance various aspects of the users' training.

If none of the functions from the list are triggered, indicating that the user's query is not related to any specific function within the course recommendation, O*NET database, and ESCO database modules, the GPT open-domain module is activated. This module leverages GPT capabilities to generate a response appropriate to the user's inquiry.

The following subsections will offer a comprehensive breakdown of each module housed within these respective blocks.

### A. PLUG-IN INTERFACE

The Plug-in Interface is powered by NGINX,[28] a high-performance web server that also serves as a reverse proxy

[28] https://www.nginx.com

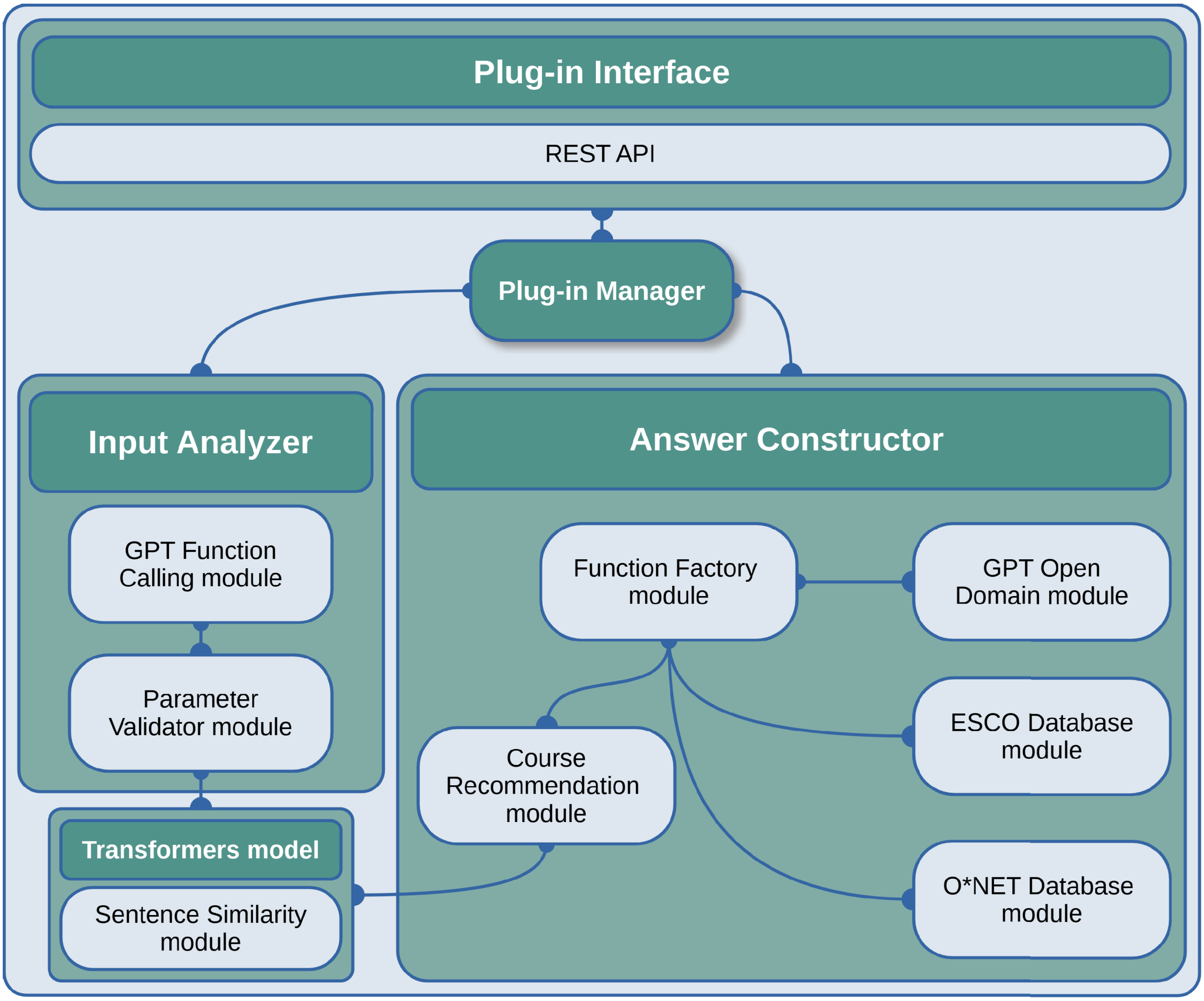



**FIGURE 1.** Knowledge plug-in architecture.

server. NGINX excels in efficiently managing concurrent connections and handling incoming requests. It operates on an event-driven, asynchronous architecture, delivering robust performance. In conjunction with NGINX, the system employs Uvicorn,[29] an ASGI web server. Uvicorn is integrated into the NGINX setup, operating as an application server to handle dynamic requests. This combination ensures a streamlined and high-performance web-serving environment, allowing for the effective processing of user requests and responses.

The REST APIs are built on FastAPI,[30] a web framework designed for constructing APIs with Python, fully adhering to open standards for APIs (OpenAPI) and JSON Schema. This framework automatically generates API documentation and offers a graphical user interface for testing purposes. The REST API module receives user input through a GET call and forwards it to the Plug-in Manager for further processing.

The results from the Answer Constructor, transmitted via the Plug-in Manager, are provided in a JSON format. This format includes the original user request, the elements queried in the databases along with their corresponding results, the answer, and additional information relevant to the identified type of request as determined by the system.

Regarding the Plug-in Manager, it accepts the input text from the REST API module and interacts with the GPT Function Calling module within the Input Analyzer block. More precisely, it sends the received input text to the GPT Function Calling module and, in return, obtains the selected function if one from the specified list is triggered, along with the extracted parameters. Afterward, it conveys these elements to the Function Factory Module to commence the process of generating a response for the user, employing the relevant function based on the nature of the question. Each function within the module gathers the necessary information

[29]https://www.uvicorn.org/
[30]https://fastapi.tiangolo.com/

for a recognized question type, formulates a response in natural language, and then returns it to the calling module.

### B. INPUT ANALYZER

The Input Analyzer block includes two modules: GPT Function Calling, and Parameter Validator. The GPT function calling module serves as a crucial component within the broader architecture of our proposed knowledge plug-in. Operating within the Input Analyzer block, the GPT function calling module employs the capabilities of the GPT model to identify patterns and extract relevant information from the user's input.[31] Upon receiving input text, the GPT Function Calling module leverages the pre-trained knowledge embedded in the GPT 3.5 model to identify specific functions listed in the following. If the input triggers a function, the method not only identifies the function but also extracts the associated parameters essential for further processing. ESCO and O*NET treat job-related skills slightly differently. In O*NET, skills are divided into technical skills, tasks, tools, skills, knowledge, and tools, whereas in ESCO there is no such distinction. Therefore, some of the following functions differ slightly in their parameter definitions.

The functions were created after we conducted interviews with both academics and industry professionals, presenting them with data from O*NET and ESCO and asking them to formulate all possible questions based on that data. This process involved 2 professors and 4 employees from two different companies. From these interactions, we compiled a comprehensive list of unique questions. To ensure accuracy and completeness, we presented this list back to the interviewees for verification and to merge questions expressed differently but sharing the same semantics. We observed that each question could be answered by extracting data from the selected tables using a limited number of SQL queries. Each function we defined therefore responds to a different question.

The list of functions together with their parameters and descriptions, is defined as follows:

1) **Function name**: alternative_names - **parameters**: the name of the job and the language of the query - **description**: list a set of alternative names or synonymous for a certain job.
2) **Function name**: abilities_for_job - **parameters**: the name of the job, the language of the query, the number of the abilities required (optional), the ability required and the type of the ability (if we want to interact with O*NET) - **description**: return a set of abilities useful for the specified job.
3) **Function name**: describe_job - **parameters**: the name of the job and the language of the query - **description**: write a description of the job.
4) **Function name**: compare_job - **parameters**: the name of the jobs to compare and the language of the query - **description**: check the common points and the differences between two jobs.
5) **Function name**: job_requiring_abilities - **parameters**: the language of the query, the number of the abilities required (optional), the ability required, and the type of the ability (if we want to interact white O*NET) - **description**: return jobs which require a certain ability activity.
6) **Function name**: teaching_something - **parameters**: the ability required and the language of the query - **description**: shows how to improve about the ability required,

After the GPT Function Calling module extracts parameters from the user's input, the Parameter Validator validates them. This involves cross-referencing the extracted parameters with the labor domain databases O*NET and ESCO, to ensure their accuracy and relevance. The search process entails computing semantic similarity between the parameters and key elements within the two databases using the Semantic Similarity module located in the Transformers model block.

By validating the parameters, this module contributes to the robustness and reliability of the knowledge plug-in, ensuring that the subsequent stages of information retrieval and response generation are based on accurate and verified input.

### C. ANSWER CONSTRUCTOR

The Answer Constructor block includes five modules: Function Factory, Course Recommendation, O*NET Database, ESCO Database, and GPT Open Domain. It is responsible for formulating responses to user queries based on the information gathered and processed by earlier stages.

The **Function Factory module** is a central element that orchestrates the process of constructing responses. It contains various functions corresponding to distinct types of questions within the labor domain and beyond. When a user's query triggers a specific function (by employing the function calling capabilities), the Function Factory module coordinates the retrieval of relevant information and the generation of a tailored response.

The **Course Recommendation module** is dedicated to suggesting courses that can enhance the user's training or skills. It accesses a database of courses, identifying the most relevant by evaluating the semantic similarity with key elements pertinent to courses within the user's input or query. The course database encompasses a selection of 180 pertinent courses, meticulously curated within the STAR project. Noteworthy contributors include *'Udemy'*, *'Coursera'*, *'EdX'*, *'FutureLearn'*, *'SimpliLearn'*, and *'Stanford'*, among others. These offerings are further augmented by search results retrieved directly from the Udemy database via its APIs.[32]

[31] https://platform.openai.com/docs/guides/function-calling

[32] https://www.udemy.com/developers/affiliate/

The **O*NET Database and ESCO Database modules** interact with O*NET and ESCO, which contain extensive information related to jobs, skills, tools, tasks, and work activities. The O*NET Database and ESCO Database modules retrieve and organize data from these sources to address user queries within the labor domain.

In scenarios where none of the predefined functions is triggered, suggesting that the user's query may be outside the specified functions, the **GPT Open Domain module** comes into play. It utilizes the capabilities of a GPT model to generate responses for more open-ended or unspecified inquiries.

As mentioned in Section I, the proposed plug-in supports multiple languages besides English. As highlighted above, among the parameters defined for each function is the *'language'* parameter. The answer generated in the **Answer Constructor module** is translated into the language specified in the parameter (if it is not English).

### D. TRANSFORMERS MODEL

The Transformers model block includes the Sentence Similarity module. We have adopted a transformers model to assess sentence similarity for computing the score between two strings, a functionality utilized by both the Input Analyzer and Answer Constructor blocks. Specifically, we employed the SentenceTransformers framework[33] [26], which facilitates easy access to BERT-based models and their derivatives such as RoBERTa [27], MPNet [28], and ALBERT [29]. These models transform input text into embedding representations, allowing for the computation of similarity, where similar embeddings correspond to pieces of text with similar meanings.

To incorporate semantic information into the embedding representations, these models undergo training using a deep model structure that includes two branches with identical deep learning layers. By feeding these branches with sentences that are syntactically distinct but convey the same or similar meaning, the model generates embeddings that encapsulate text semantics. Such models stand as state-of-the-art technology for addressing Semantic Textual Similarity (STS) tasks. Within the proposed methodology, we utilized the model 'all-MiniLM-L6-v2'.[34] to compute text similarities.

This model underwent initial training through a distillation approach, involving the transfer of knowledge from a teacher model to a student model. Subsequently, it underwent fine-tuning on 1 billion sentence pairs using a contrastive learning approach, where, given an input sentence, the model learned to identify the most similar one from a set of possible alternatives. The selection of this model is based on its compact size, speed, and commendable performance across various benchmarks.[35]

[33] https://www.sbert.net/
[34] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[35] https://www.sbert.net/docs/pretrained_models.html

## V. OUR SYSTEM AT WORK

Figure 2 illustrates all possible sequences for the proposed plug-in. To focus on the most significant scenarios, we envisioned three possibilities:

1) In the first scenario, the user's input does not trigger any of the predefined functions.
2) The second scenario outlines a situation where the activated function requires data from one of the labor databases (ESCO and O*NET).
3) The third scenario portrays a situation involving the course database.

In the following paragraph, we will present an example of the three cases and demonstrate how the modules depicted in Figure 2 interact with each other in each of the illustrated instances.

In the first scenario, an example of user query *"When was the web born?"* would fail to activate any predefined functions within the GPT Function Calling module. Consequently, the module notifies the Plug-in Manager that no function was activated. Subsequently, the Plug-in Manager directs the user input to the Function Factory Module located within the Answer Construction block. This module utilizes the GPT Open Domain module, which leverages the capabilities of the GPT model, to formulate a response to the user query. The resultant answer is then relayed back to the Plug-in Interface for transmission to the user.

In scenario 2, for instance, when the user's input *"Describe the job of a taxi driver"* is received and the O*NET source database or the ESCO source database is chosen by the user from the Plug-in interface, the *"describe_job"* function is activated.

The system extracts *"taxi driver"* as the job_name parameter and identifies *"English"* as the language from the query. Subsequently, it directs the former to the Parameter Validator module, as depicted in Figure 3.

This parameter is then encoded by the Transformers module, and its embedding is compared with entries in the database. The most similar entry in the database is identified as *"Taxi Drivers"* (Figure 4).

The Plug-in Manager then sends a request to the Function Factory, providing the function name (*"describe_job"*), the validated parameter (*"Taxi Drivers"*), the selected database (O*NET), and the user's language (*"English"*). Subsequently, the Function Factory employs the O*NET Database module to retrieve the requested information about the job description of the *"Taxi Drivers"* occupation class, which is: *"Drive a motor vehicle to transport passengers on an unplanned basis and charge a fare, usually based on a meter."* Finally, the generated answer is returned to the Plug-in Interface for delivery to the user (Figure 5).

If the chosen database is ESCO, the flow is the same. An example of a response generated from the ESCO database information with the same query is shown in Figure 6.

Finally, an example for scenario 3, upon receiving the user query *"Can you teach me something about Python?"*, the system activates the *teaching_something* function. The Function

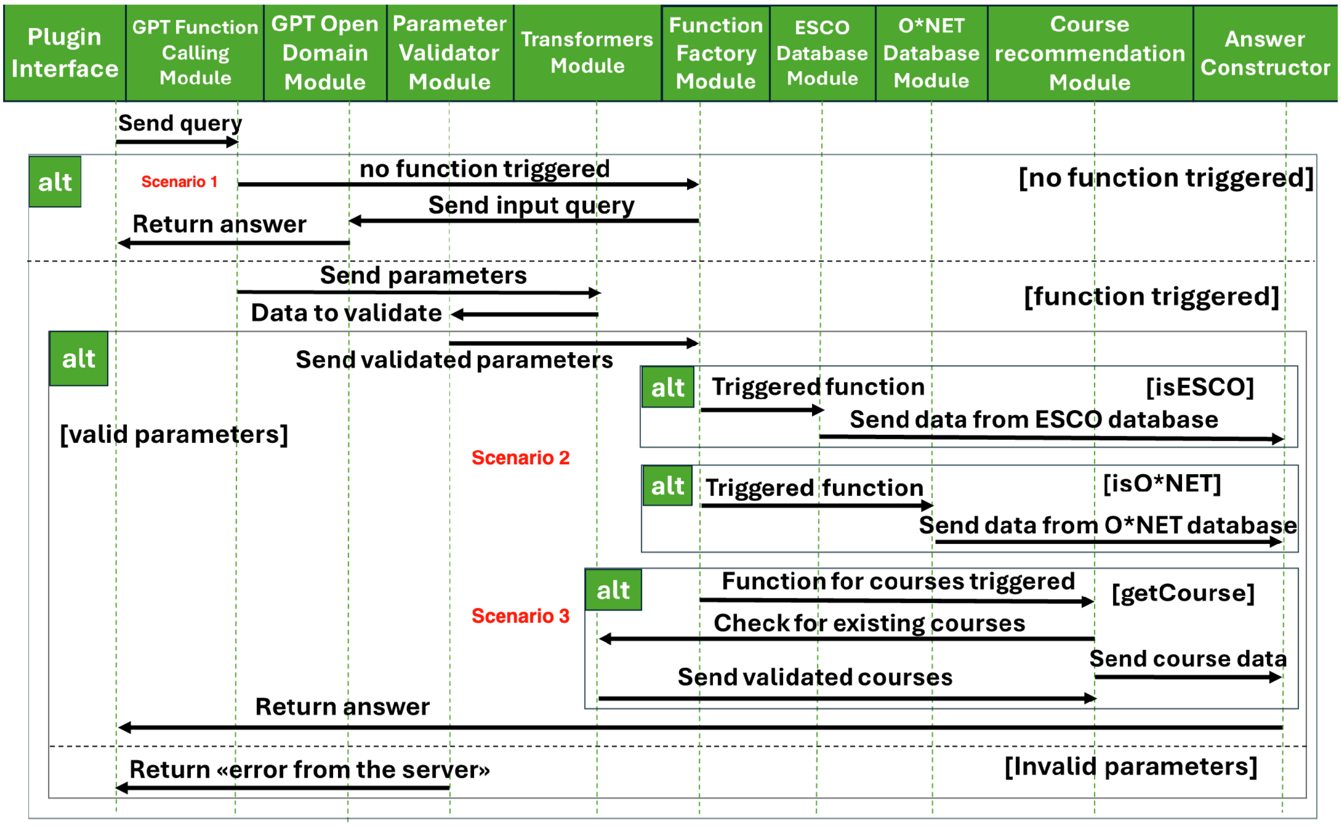



**FIGURE 2.** Interaction of modules for three possible scenarios.

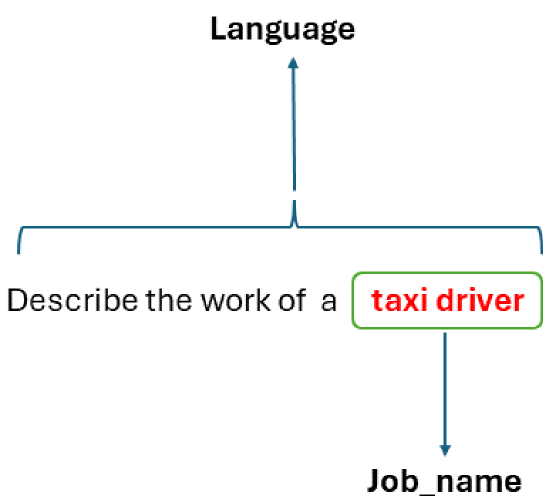



**FIGURE 3.** Parameters extraction.

Factory module then triggers the Course Recommendation module to compare the extracted parameter *"Python"* with the information within the list of available courses.

The Course Recommendation module employs the Transformers module to generate embeddings for both the user query and the list of potential candidate courses. Subsequently, through the calculation of cosine similarity, the module identifies the most relevant course. Finally, information such as the course name and link associated with the selected course is returned to the user (Figure 7).

## VI. PERFORMANCE EVALUATION

To conduct a comprehensive assessment of our tool, direct comparisons with other LLMs were deemed inappropriate. This decision was based on the distinct advantage our plug-in holds, leveraging information from two ad-hoc databases that are not available to other LLMs. Consequently, we undertook two distinct evaluations to gauge the efficacy of our approach.

The initial assessment focused on the precision of our plug-in in accurately triggering the function associated with the user's query. This metric served as an indicator of the tool's ability to understand and respond appropriately to user input.

The second evaluation centered around a user study designed to measure the responsiveness, effectiveness, and overall quality of interactions when our knowledge plug-in was integrated with ChatGPT. This comprehensive analysis aimed to provide insights into the user experience and the synergistic impact of our integration with the existing ChatGPT framework.

### A. FUNCTIONS PRECISION ASSESSMENT

For each function outlined in Section IV-B, we engaged four users from distinct companies and countries (Italy, Spain, France, United Kingdom) to formulate three questions in their

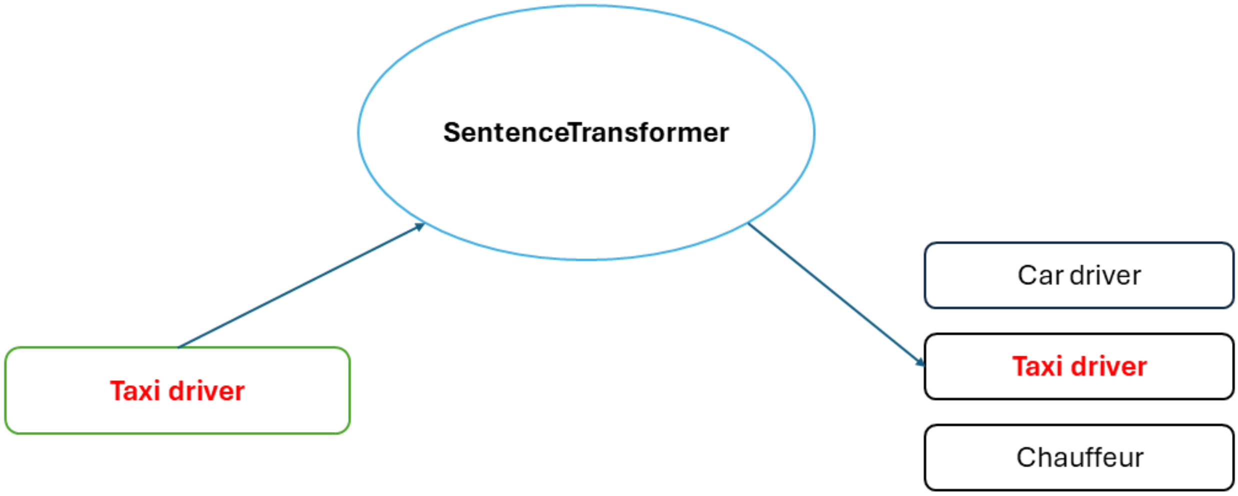



**FIGURE 4.** Comparison between embedding and entries of the database chosen.

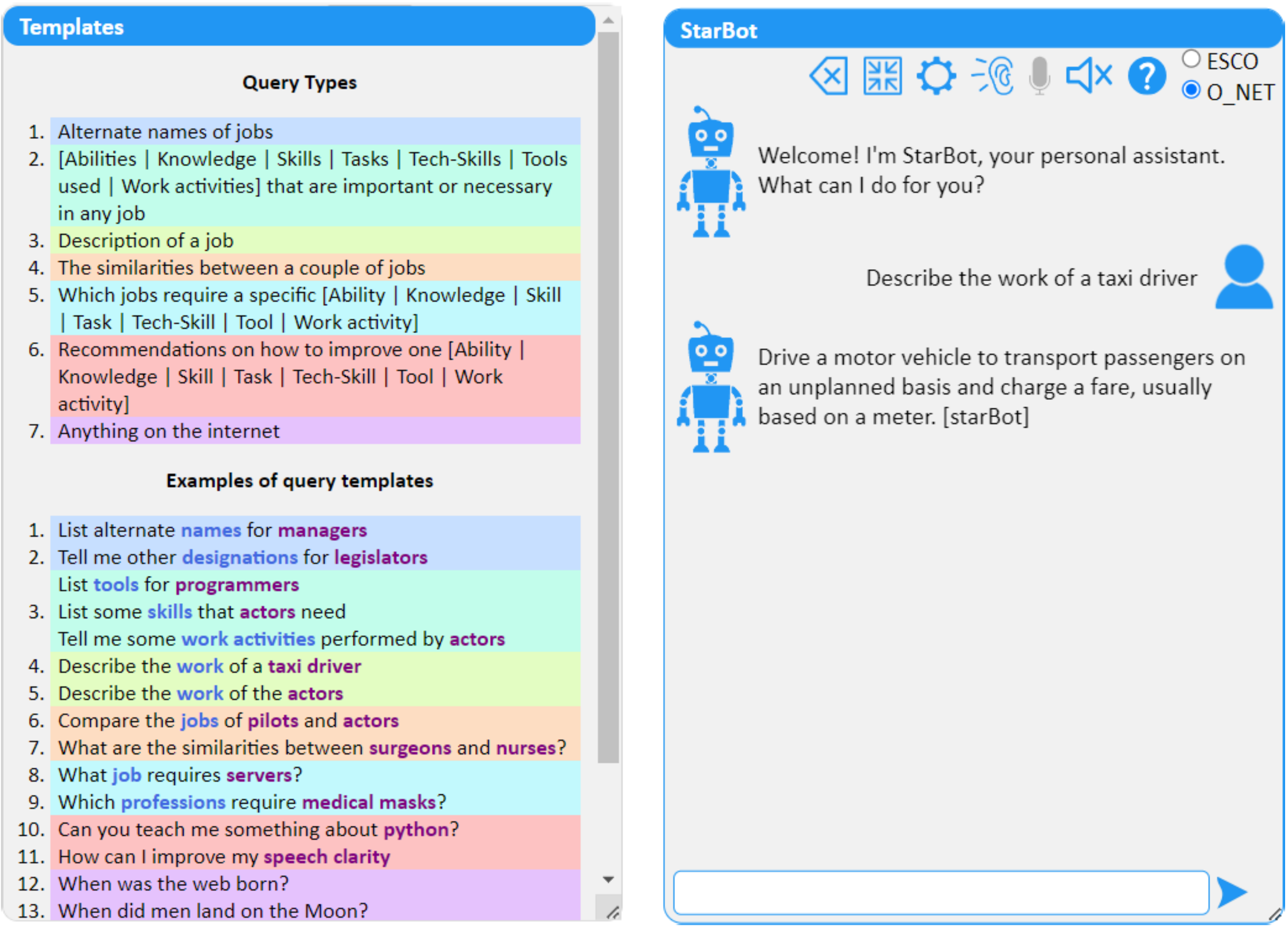



**FIGURE 5.** Generated answer shown in the UI (O*NET).

respective natural languages. This process yielded a total of 72 questions (18 questions per user). The primary objective was to validate that different versions of the same question could consistently trigger the corresponding function.

The results proved to be impressive, as each function consistently and accurately responded to its associated question variants. This remarkable outcome demonstrated a precision rate of 100%, affirming the robustness and reliability of

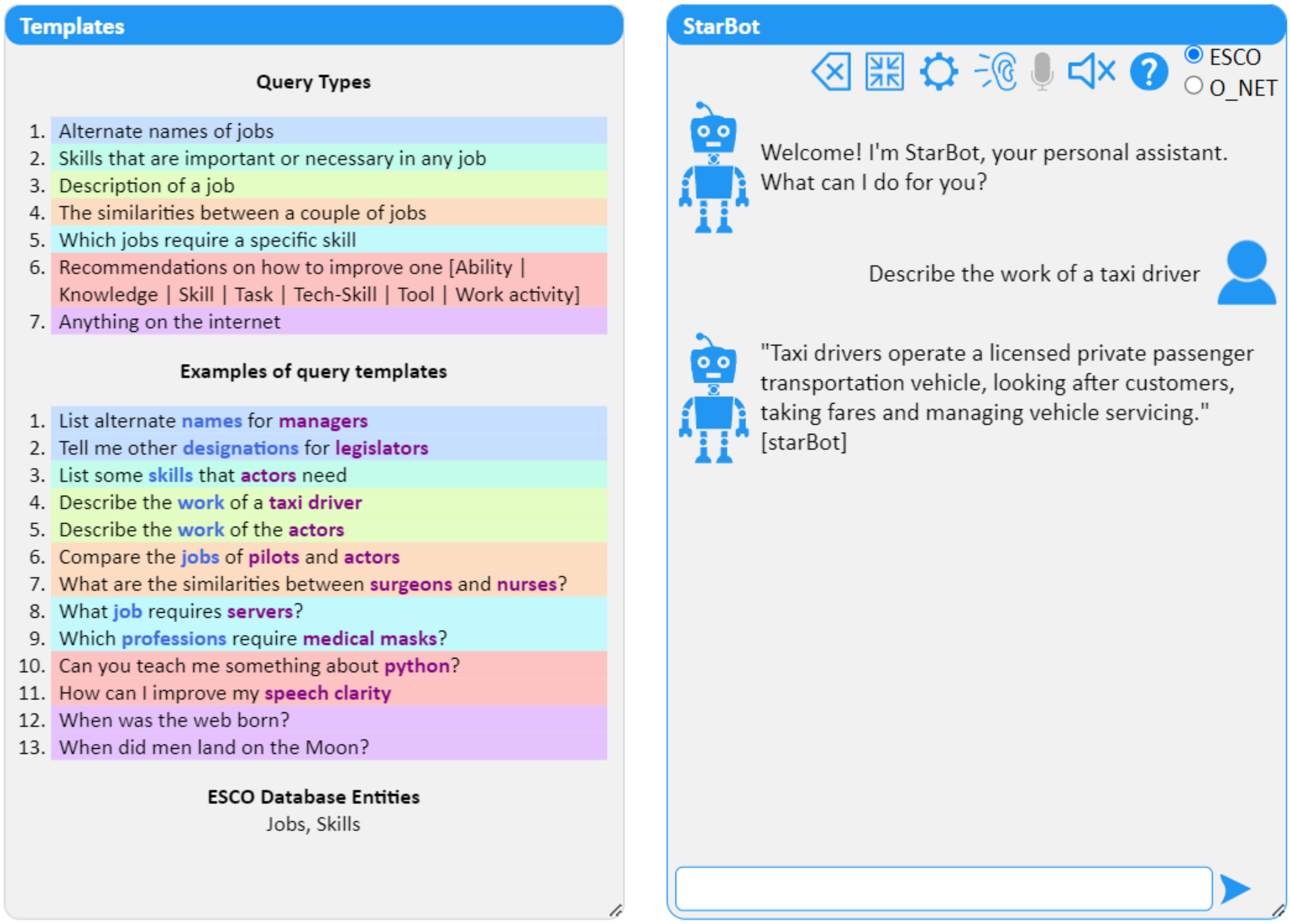



**FIGURE 6.** Generated answer shown in the UI (ESCO).

our system in successfully activating the intended functions across diverse languages. The list of the 72 questions can be found at http://192.167.149.18/A-Seamless-ChatGPT-Knowledge-Plugin-about-the-labor-market-main.zip.

### *B. USER STUDY*

We provided access to the conversational agent to 9 participants from two different companies.

Each participant received an introduction to the prototype together with instructions and highlights of its key features. Subsequently, participants were actively engaged with the prototype, followed by the completion of a comprehensive four-part survey capturing their experiences.

The initial section of the survey gathered background information on each participant. The subsequent segment featured a standard System Usability Scale (SUS) questionnaire[36] [30], designed to assess the overall usability of the proposed system. The third section comprised questions regarding the quality of interaction, the effectiveness of recommendations, and the conveyed information, utilizing a five-point Likert scale for responses. The final part consisted of five open-ended questions, encouraging participants to provide insights into the system's strengths, weaknesses, and suggestions for additional features. Detailed results from these sessions are presented in the subsequent paragraphs.

In terms of the user demographic, we selected 9 individuals with varied backgrounds. On average, these participants had accumulated 11.5 years of professional experience. Their areas of expertise covered a broad spectrum, encompassing fields such as software engineering, automation, project management, sentiment analysis, big data, semantic web, data science, AI applications, robotics, mobile development, and mobile technologies.

The score obtained from the SUS questionnaire is 81.39/100, equivalent to an A grade and positioning our prototype in the 90th percentile rank.[37] Figures 8 and 9 showcase users' responses to the SUS questionnaire. It is worth noting that while the even questions (Figure 9) are

[36]System Usability Scale (SUS) - https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

[37]Interpreting a SUS score - https://measuringu.com/interpret-sus-score/

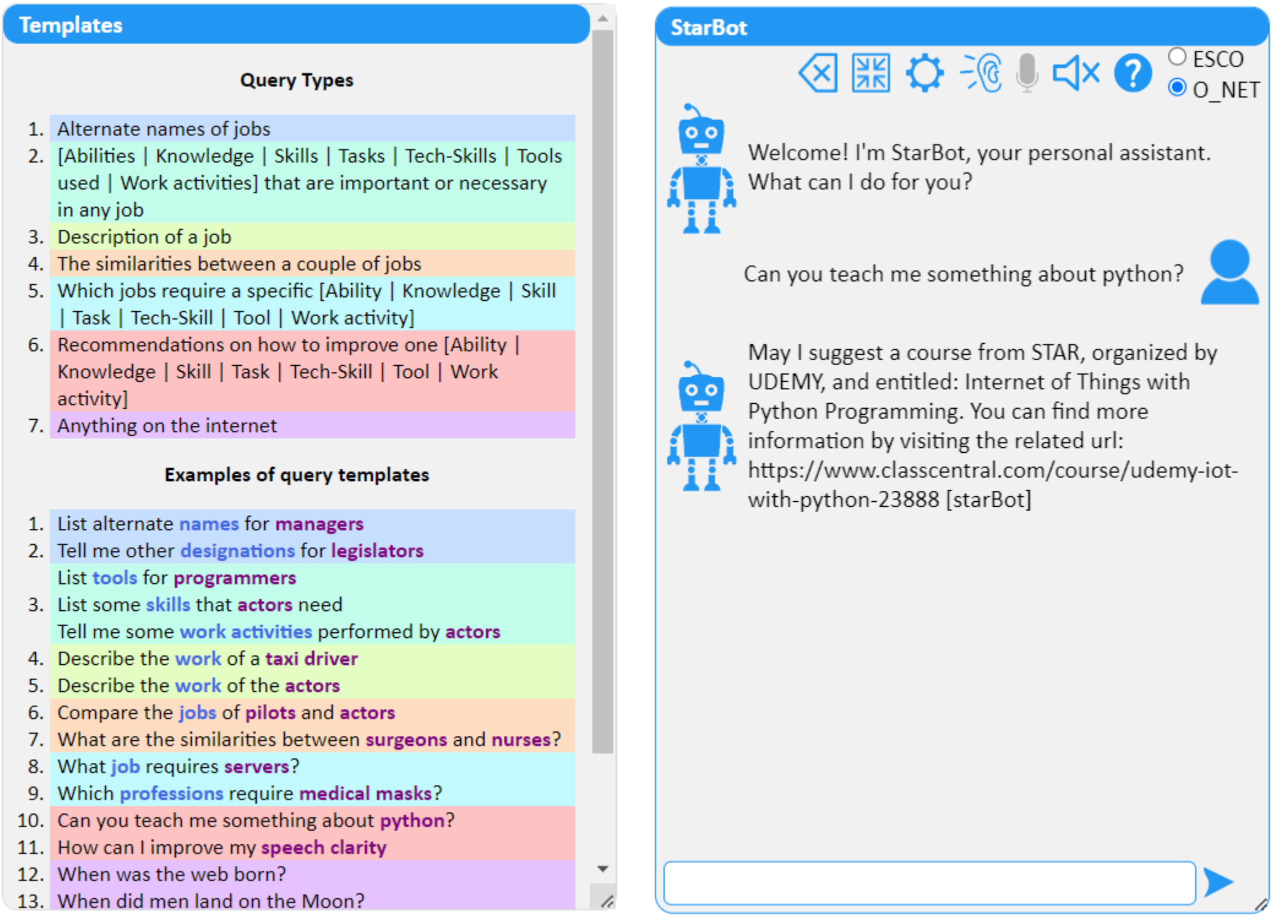



**FIGURE 7.** Generated answer shown in the UI (Course).

framed negatively (lower scores indicating better results), the odd questions (Figure 8) are framed positively (higher scores indicating better results).

In the following sections, we will present the averages and standard deviations for specific questionnaire items. Respondents expressed a high inclination for frequent system use, evidenced by an average score of $3.33 \pm 0.47$. They found the system notably uncomplicated, achieving an average score of $1.33 \pm 0.67$. Participants perceived the system as easy to use, recording an average score of $4.56 \pm 0.50$, and felt confident navigating it independently, without requiring technical assistance (average score of $1.44 \pm 0.68$).

Additionally, respondents acknowledged the well-integrated functionalities of the system, resulting in a high average score of $3.67 \pm 0.67$ and did not perceive inconsistency (average score of $1.56 \pm 0.83$). Participants anticipated a quick learning curve for the system, reflected in an average score of $4.44 \pm 0.68$, and appreciated its non-cumbersome nature (average score of $1.67 \pm 0.82$). Lastly, users reported feeling very confident using the system, as evidenced by an average score of $4.11 \pm 0.87$, and expressed that it did not necessitate extensive learning before becoming familiar with it (average score of $1.56 \pm 0.96$).

The evaluators were assigned the task of appraising the quality of interaction facilitated by the developed knowledge plug-in. Three questions were presented to them, each rated on a five-point Likert scale (where 1 indicated low quality, and 5 indicated high quality). These questions were:

1) How do you assess the quality of the interaction with the knowledge plug-in?
2) How do you rate the recommendations provided by the knowledge plug-in?
3) How do you rate the provision of information from the knowledge plug-in about a certain job?

The average and standard deviation of their responses are outlined in Table 1.

Additionally, for each of the three aforementioned questions, annotators were encouraged to provide feedback. This feedback focused on the quality of interaction with the plug-in, the efficacy of recommendations, and the information provided for each job.

Two users raised concerns regarding the GUI, pointing out that three panels overlap initially. They suggested addressing this issue to ensure non-overlapping panels. Additionally, one user noted difficulty using the system for individuals lacking ICT skills. Conversely, another user expressed enthusiasm

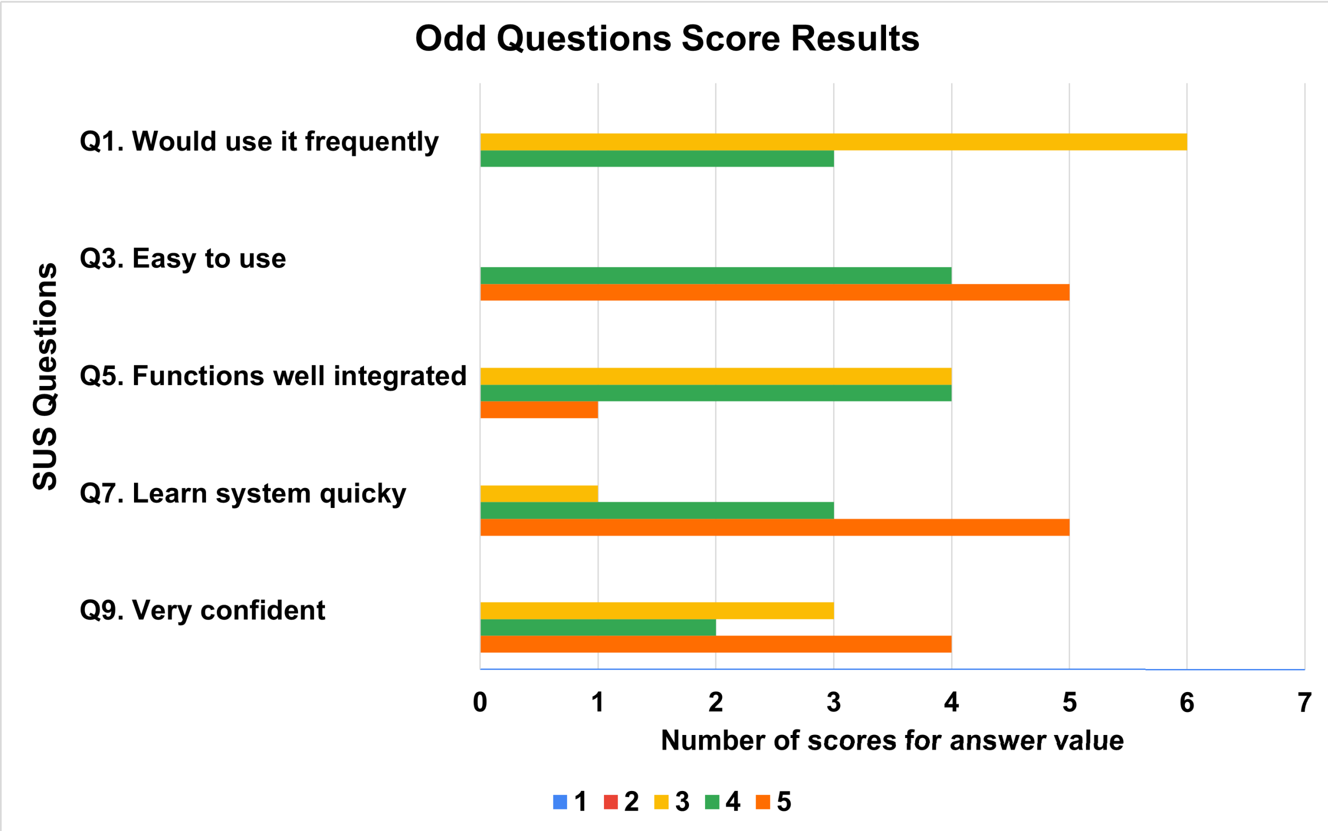



**FIGURE 8.** SUS questionnaire results (Positive Questions).

**TABLE 1.** Quality assessment questions.

| Question | Average | Standard deviation |
|---|---|---|
| How do you assess the quality of the interaction with the knowledge plug-in? | 3.89 | ±0.74 |
| How do you rate the recommendations provided by the knowledge plug-in? | 4.33 | ±0.67 |
| How do you rate the provision of information from the knowledge plug-in about a certain job? | 3.44 | ±1.17 |

for the GUI, describing it as simple and highly practical. This user recommended displaying the hierarchy panel only when dealing with very generic questions. Lastly, a user commended the system's result quality, expressing overall satisfaction.

Regarding recommendations provided by the chatbot, users were generally satisfied. One user advised including the link to the suggested training course, while another suggested incorporating an example or image to illustrate the application of job skills. A third user pointed out the absence of context from previous interactions.

Concerning information provision, one user noted that the returned information perfectly aligns with the database, and no responses with data not present there were observed. However, another user suggested organizing responses with skills ordered by importance instead of alphabetical order. A third user expressed the desire to view all job skills, beyond just the initial five, perhaps by including a link to access the complete list. Finally, a user raised a complaint regarding the data contained in the database.

### C. OPEN QUESTIONS

This section will delve into the feedback from users in response to the open-ended questions. It is essential to acknowledge that not all users responded to every question.

#### 1) Q1. WHAT ARE THE MAIN STRENGTHS OF THE KNOWLEDGE PLUG-IN?

Five users highlighted the flexibility and user-friendly interactions with the system, emphasizing its capability to provide relevant and trustworthy data. Another user appreciated the option to choose between the ESCO and O*NET databases for job-related information. A final user was impressed by the knowledge plug-in's ability to offer highly specialized advice on occupational information. Furthermore, they pointed out that the system's transparency about responses coming

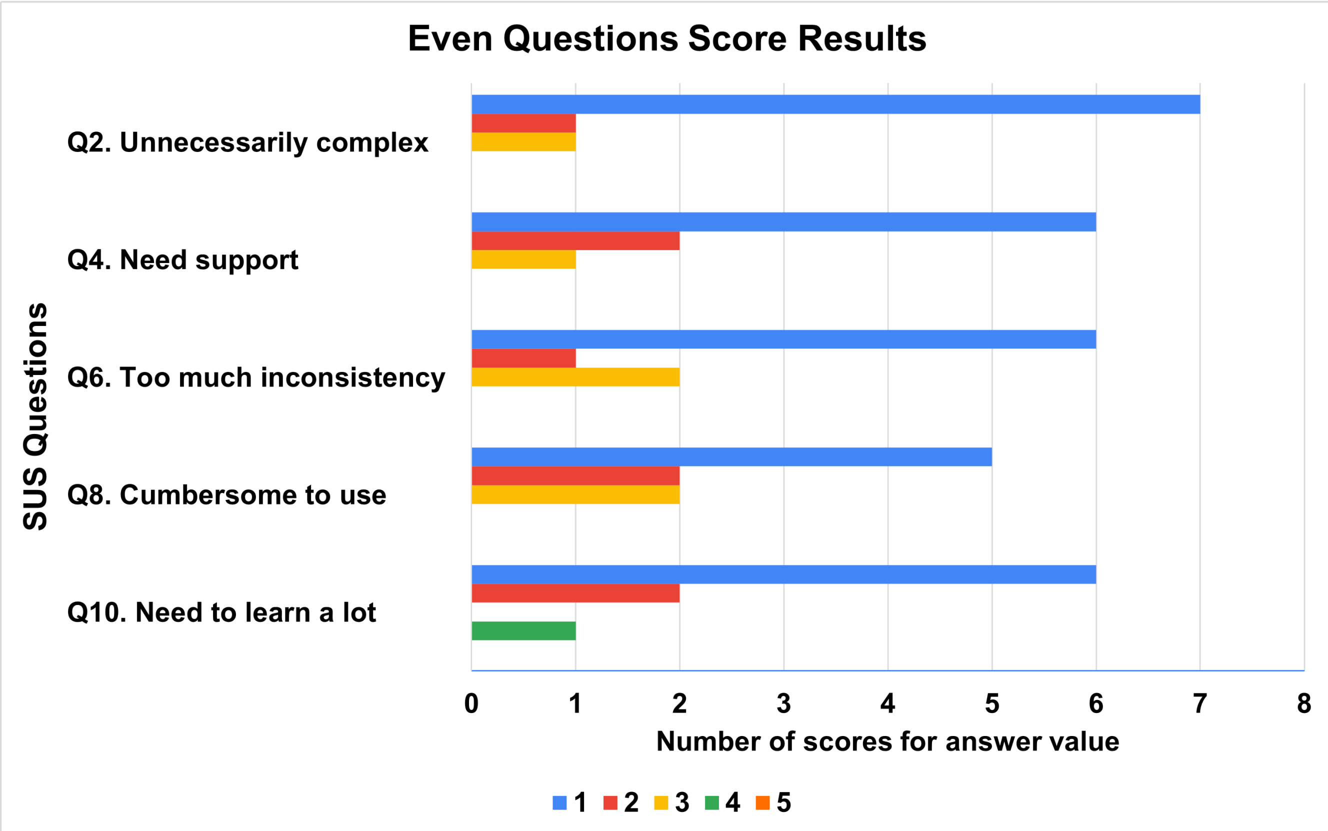



**FIGURE 9.** SUS questionnaire results (Negative Questions).

from reliable databases enhances trust and encourages practical use.

#### 2) Q2. WHAT ARE THE MAIN WEAKNESSES OF THE KNOWLEDGE PLUG-IN?

One user noted occasional delays in the system's response time. Another expressed dissatisfaction with the lack of detailed responses and highlighted the absence of context in the returned information. Additionally, a user acknowledged the system's capability to provide high-quality responses but mentioned its occasional difficulty in recognizing slightly varied sentences. Furthermore, the system's inability to remember context was pointed out, posing a potential hindrance to seamless conversation flow. Lastly, a user voiced concerns about the absence of a tutorial on how to use the tool.

#### 3) Q3. CAN YOU THINK OF ANY ADDITIONAL FEATURES TO BE INCLUDED IN THE KNOWLEDGE PLUG-IN?

Users offered diverse suggestions, such as incorporating follow-up questions for general queries to enable more specific and detailed answers. Another user proposed the inclusion of links to additional information sources to enhance the comprehensiveness of the responses. Additionally, two users recommended integrating more visual elements, such as images, and improving formatting with features like bullet lists for better visualization. The inclusion of navigable links to training courses and additional text in the hierarchy window was also suggested as a valuable enhancement.

#### 4) Q4. CAN YOU THINK OF ANY ADDITIONAL TYPES OF QUERIES FOR THE KNOWLEDGE PLUG-IN?

One user expressed the desire to inquire about links to job portals or other relevant reading materials. Additionally, two users expressed interest in querying statistical data about a specific occupation in various countries. Another user wishes to ask about describing a skill and how it can be developed. Finally, one more user would like to inquire about the recommended course of study for a specific job and the number of years of experience required.

#### 5) Q5. WHAT WOULD YOU ADD TO INCREASE THE ACCURACY/COMPREHENSIVENESS OF THE INFORMATION RETURNED BY THE KNOWLEDGE PLUG-IN?

Users provided the following suggestions:

- Introduce different intents for the same question.
- Explore the implementation of a voting system: users can evaluate the bot's responses, and these marks can serve as weights to determine whether to provide a specific answer or defer to ChatGPT. If a response receives negative votes and there's no alternative in the database, ChatGPT's answer will be prioritized.

- Retain the context of previous messages for a more coherent conversation.
- Incorporate the ability to gather user feedback on bot responses. Even if not currently utilized, it can be valuable for data collection.

## VII. DISCUSSION

We are aware of the limited sample size used for the performance validation. Yet, it is worth noting that these preliminary results are in general positive, and that received critiques are constructive and worth full attention. For the discussion of results, we decided to separately comment on the SUS scores and the answers to open questions, rather than attempting a unified analysis. Indeed, hard data from SUS will clearly define the state-of-art of our Knowledge Plug-in, confirming or not that the research is on the right path. On the other hand, open answers may add a more nuanced evaluation, to guide our future course of action.

The exam of SUS data for this preliminary sample highlights a few important points:

- *User friendliness*. The evaluators praised the system for being easy to use (Q3 and Q4) with a smooth learning curve (Q7 and Q10) and for its simplicity expressed as lack of complexity (Q2) and not being cumbersome (Q8). This confirms our careful choices in the user interface (UI) and the user experience (UX), both needed to create a user-friendly system [31].
- *Solidity*. With high scores in the dimensions of confidence (Q9) and integration (Q5) as well as a general lack of inconsistency (Q6), our Knowledge Plug-in transmitted to the validators an overall idea of a solid and self-coherent system. This is once again a confirmation of our design choices for the UX.

We may want to reflect on the Q1 score: validators claimed not to be very prone to often use the system. Considering SUS scores, this is surely not due to usage difficulties or lack of trust. Rather, this answer suggests to increase the perceived usefulness of the tool. And here is where the open-answer questions proved useful.

First, it is important to observe how answers confirm the results of SUS. Appreciation of flexibility, trustworthiness, and adoption of authoritative sources prove the solidity and user-friendliness of the Knowledge Plug-in. Yet, the critiques are even more interesting, because most of them point to the same, precise request: *improve the answers*. In fact, our validators suggested:

- to enrich the information provided – with more details, statistical data, and better descriptions.
- To add context to the answers, allowing users to weave a conversation rather than having a one-time question answering.

We also found the suggestion to implement a voting system whose ultimate effect is to decide the source for a given answer (the databases or GPT) interesting. Such a non-trivial functionality would introduce an important level of complexity. At the same time, it would also put our work in a mainstream tendency towards "closed-loop" systems, where end users challenge authoritative sources and demand to consider their opinions - be it in re-ranking the results on a search engine, or in a work-related plug-in.

## VIII. CONCLUSION AND FUTURE WORK

In this article, we presented a Knowledge Plug-in designed to give relevant information related to the complex world of work. With an easy-to-use and easy-to-learn interface, our tool can deliver quality information on skills, competencies, and jobs, extracted from authoritative, international sources. Integration with ChatGPT APIs empowers the Plug-in, giving it the ability to go beyond job-related questions and enrich the user experience. To assess our work, we conducted a validation with selected reviewers. While the level of appreciation and acceptance is encouragingly high, we also collected constructive criticism and ideas that may guide our future work. Based on reviewers' feedback, the next areas of investigation could be:

- To improve the answers that the Knowledge Plug-in delivers. Today, the user gets only the information available in the international databases, often a very concise text, not necessarily easy to understand. The usage of authoritative sources is unavoidable to produce certified and reliable answers. Yet, we may envisage the use of generative AI to enrich standard information, provide examples, and offer explanations.
- To integrate answers with additional materials, like statistical data related to a certain job, or visual information and links to complement text.
- To evolve *from question answering to conversation*, considering context and possibly user preferences, to deliver a vibrant, engaging user experience. This may involve an extensive integration of generative AI in the tool.
- To evaluate user feedback collection, eventually implementing the voting system described above. This is a complex work requiring a considerable amount of research, not only for the implementation but also for the need to weigh the pros and contras of challenging an authoritative database. It is relevant that the authors are already researching human-based re-ranking of AI results, thanks to a grant from Next Generation Internet initiative.[38]

Another aspect where the Knowledge Plug-in may improve is the handling of multilanguage. There is strong evidence that language barriers are an increasing challenge in the world of work, especially in segments like construction.[39] While adoption of ESCO database, available in all the EU languages, is a significant step forward, we may explore the possibility of going beyond, for example either integrating the ISCO database or even combining authoritative sources with generative AI, to also cover non-European languages.

[38] https://www.ngi.eu/about/
[39] https://www.languageconnectsfoundation.org/uploads/files/general/MakingLanguagesOurBusiness_FullReport.pdf

We feel that the core of our future research will be the attempt to get a delicate integration of dependable, certified, and authoritative sources with engaging, rich, and flexible information coming from generative AI. What is at stake is the creation of methodologies and tools that, leveraging the most advanced technologies, may bring to a more inclusive world of work.

## REFERENCES

[1] S. M. Suhaili, N. Salim, and M. N. Jambli, "Service chatbots: A systematic review," *Expert Syst. Appl.*, vol. 184, Dec. 2021, Art. no. 115461.

[2] A. Meloni, S. Angioni, A. Salatino, F. Osborne, D. R. Recupero, and E. Motta, "Integrating conversational agents and knowledge graphs within the scholarly domain," *IEEE Access*, vol. 11, pp. 22468–22489, 2023.

[3] J. Dhameliya and N. Desai, "Job recommender systems: A survey," in *Proc. Innov. Power Adv. Comput. Technol. (i-PACT)*, vol. 1, Mar. 2019, pp. 1–5.

[4] J. Jarrett and M. B. Blake, "Using collaborative filtering to automate worker-job recommendations for crowdsourcing services," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2016, pp. 641–645.

[5] W. Shalaby, B. AlAila, M. Korayem, L. Pournajaf, K. AlJadda, S. Quinn, and W. Zadrozny, "Help me find a job: A graph-based approach for job recommendation at scale," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 1544–1553.

[6] A. Giabelli, L. Malandri, F. Mercorio, M. Mezzanzanica, and A. Seveso, "Skills2Job: A recommender system that encodes job offer embeddings on graph databases," *Appl. Soft Comput.*, vol. 101, Mar. 2021, Art. no. 107049.

[7] Y. Lu, S. El Helou, and D. Gillet, "A recommender system for job seeking and recruiting website," in *Proc. 22nd Int. Conf. World Wide Web*, May 2013, pp. 963–966.

[8] R. Mishra and S. Rathi, "Enhanced DSSM (deep semantic structure modelling) technique for job recommendation," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7790–7802, Oct. 2022.

[9] C. Zhu, H. Zhu, H. Xiong, C. Ma, F. Xie, P. Ding, and P. Li, "Person-job fit: Adapting the right talent for the right job with joint representation learning," *ACM Trans. Manage. Inf. Syst.*, vol. 9, no. 3, pp. 1–17, Sep. 2018.

[10] C. Qin, H. Zhu, T. Xu, C. Zhu, L. Jiang, E. Chen, and H. Xiong, "Enhancing person-job fit for talent recruitment: An ability-aware neural network approach," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 25–34.

[11] R. Ramanath, H. Inan, G. Polatkan, B. Hu, Q. Guo, C. Ozcaglar, X. Wu, K. Kenthapadi, and S. C. Geyik, "Towards deep and representation learning for talent search at LinkedIn," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 2253–2261.

[12] C. Baik, H. V. Jagadish, and Y. Li, "Bridging the semantic gap with SQL query logs in natural language interfaces to databases," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 374–385.

[13] S. Abbas, M. U. Khan, S. U. Lee, A. Abbas, and A. K. Bashir, "A review of NLIDB with deep learning: Findings, challenges and open issues," *IEEE Access*, vol. 10, pp. 14927–14945, 2022.

[14] X. V. Lin, R. Socher, and C. Xiong, "Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing," 2020, *arXiv:2012.12627*.

[15] T. Scholak, N. Schucher, and D. Bahdanau, "PICARD: Parsing incrementally for constrained auto-regressive decoding from language models," 2021, *arXiv:2109.05093*.

[16] T. Lee, T. Zhu, S. Liu, L. Trac, Z. Huang, and Y. Chen, "CASExplorer: A conversational academic and career advisor for college students," in *Proc. 9th Int. Symp. Chin. (CHI)*, vol. 4, Oct. 2021, pp. 112–116.

[17] M. X. Zhou, G. Mark, J. Li, and H. Yang, "Trusting virtual agents: The effect of personality," *ACM Trans. Interact. Intell. Syst.*, vol. 9, nos. 2–3, pp. 1–36, Sep. 2019.

[18] G. M. Biancofiore, T. Di Noia, E. Di Sciascio, F. Narducci, and P. Pastore, "GUApp: A knowledge-aware conversational agent for job recommendation," in *Proc. Joint KaRS ComplexRec Workshop (CEUR-WS)*, 2021, pp. 1–6.

[19] S. B. Zinjad, A. Bhattacharjee, A. Bhilegaonkar, and H. Liu, "Resume-Flow: An LLM-facilitated pipeline for personalized resume generation and refinement," 2024, *arXiv:2402.06221*.

[20] T. Alqahtani, H. A. Badreldin, M. Alrashed, A. I. Alshaya, S. S. Alghamdi, K. bin Saleh, S. A. Alowais, O. A. Alshaya, I. Rahman, M. S. Al Yami, and A. M. Albekairy, "The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research," *Res. Social Administ. Pharmacy*, vol. 19, no. 8, pp. 1236–1242, Aug. 2023.

[21] M. S. Iswahyudi, N. Nofirman, I. K. A. Wirayasa, S. Suharni, and I. Soegiarto, "Use of ChatGPT as a decision support tool in human resource management," *Jurnal Minfo Polgan*, vol. 12, no. 1, pp. 1522–1532, Aug. 2023.

[22] G. Sebastian, "Hello! this is your new HR assistant, ChatGPT! impact of AI chatbots on human resources: A transformative analysis," May 2023, doi: 10.2139/ssrn.4461474. [Online]. Available: https://ssrn.com/abstract=4461474

[23] N. Ho Koh, J. Plata, and J. Chai, "BAD: BiAs detection for large language models in the context of candidate screening," 2023, *arXiv:2305.10407*.

[24] *The Crosswalk Between ESCO and O*Net (Technical Report)*, ESCO Publications, Eur. Commission, ESCO, St. Louis, MI, USA, 2022.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[26] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," 2019, *arXiv:1908.10084*.

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

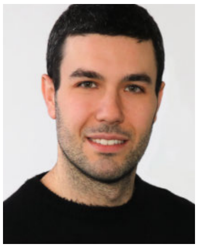
[28] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "MPNet: Masked and permuted pre-training for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 16857–16867.

[29] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.

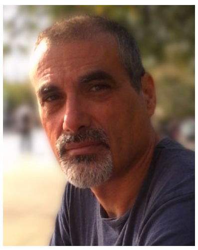
[30] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale,'" in *Industry*, vol. 189, 1996. P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland, Eds. London, U.K.: Taylor & Francis, 1996.

[31] D. A. Norman, *The Psychology of Everyday Things*. New York, NY, USA: Basic Books, 1988.

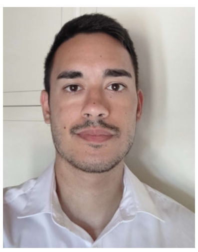
**RUBÉN ALONSO** received the Diploma degree in advanced studies on computing systems from the University of the Basque Country, Spain, and the degree in computer science and engineering from the University of Deusto, Spain. Since 2004, he has been involved in industrial and research projects, mainly related to computational trust, security, and embedded systems. He was part of the Trust and Security Unit, European Software Institute, and was responsible for the security and embedded systems group of visual tools. He was also a Visiting Researcher with the VTT Technical Research Centre of Finland and the University of Trento. Currently, he coordinates the ICT and Robotics Department, R2M Solution. His research interests include Linux embedded systems, robotics, IT security, and data mashups. He has experience in the European Research Programmes as well as industrial experience with various firms.



**DANILO DESSÍ** received the master's and Ph.D. degrees from the University of Cagliari, Italy. Since August 2024, he has been an Assistant Professor at the Department of Computer Science, College of Computing and Informatics of the University of Sharjah. From October 2022 to July 2024, he was a Senior Researcher at the KTS Department, GESIS Leibniz Institute for the Social Sciences, since October 2022. Previously, he was a Researcher at the University of Cagliari, and a Post-Doctoral Researcher/Senior Researcher at FIZ Karlsruhe – Leibniz Institute for Information Infrastructure and Karlsruhe Institute of Technology (KIT). He has been a Visiting Researchers in the following centers all around the world: Philips Research (Eindhoven, 2016), Center for Data Science NYU (New York City, 2017), Knowledge Media Institute — The Open University (Milton Keynes, 2018), University of Paris 13 (Paris, 2019). He is a co-author of CS-KG, a large-scale automatically generated knowledge graph about computer science and the developer of its construction pipeline. He has co-authored more than 40 research publications in top-tier international conferences and journals. His current research interests include artificial intelligence, knowledge graphs, science of science, deep learning, and natural language processing.



**ANTONELLO MELONI** received the B.S. and M.S. degrees in computer science from the University of Cagliari, Italy, where he is currently pursuing the Ph.D. degree. His research interests include natural language processing, human–robot interaction, conversational agents, and semantic web. He is the Key Developer of amr2fred, a tool for translating abstract meaning representation to motif-based linguistic knowledge graphs and a Key Developer of AIDA-bot, a conversational agent to explore scholarly knowledge graphs. Since 1998, he has been working for local public administrations as a System Administrator, an Expert in geographic information systems and CAD, a Software Developer, and an IT Infrastructure Designer.



**MARCO MURGIA** received the B.S. and M.S. degrees in computer science from the University of Cagliari, Italy, where he is currently pursuing the Ph.D. degree. For his master's thesis, he focused on citation prediction using generative models, such as GPT-2 and Named Entity Recognition (NER) with BERT, which culminated in the publication titled "Citation Prediction by Leveraging Transformers and Natural Language Processing Heuristics." At the beginning of his Ph.D., he has been collaborating with R2M Solution (a company that co-founded his fellowship), contributing to AI applications within the labor market as part of the European project STAR. His research interests include natural language processing (NLP), large language models, and citation prediction.

**DIEGO REFORGIATO RECUPERO** received the Ph.D. degree in computer science from the University of Naples Federico II, Italy, in 2004. He has been a Full Professor with the Department of Mathematics and Computer Science, University of Cagliari, Italy, since February 2022. From 2005 to 2008, he was a Postdoctoral Researcher with the University of Maryland College Park, USA. He won different awards in his career (such as the Marie Curie International Reintegration Grant, Marie Curie Innovative Training Network, Best Researcher Award from the University of Catania, Computer World Horizon Award, Telecom Working Capital, Startup Weekend, and Best Paper Award). He co-founded six companies within the ICT sector and is actively involved in European projects and research (with one of his companies he won more than 40 FP7 and H2020 projects). His current research interests include sentiment analysis, semantic web, natural language processing, human–robot interaction, financial technology, and smart grids. He is the author of more than 200 conference and journal articles in these research fields, with more than 2800 citations.

**GIUSEPPE SCARPI** received the B.S. degree in psychology and the M.S. degree in electronic engineering. He started working in the far 1993. Since then, he carried out a wide variety of assignments in companies of all sizes–from Leonardo and Telecom Italia to micro-companies with five employees. During his long career, he held many technical and managerial positions. Since 2022, he has been the Business Development Manager with the Innovation Division, R2M Solution.

Open Access funding provided by 'Università degli Studi di Cagliari' within the CRUI CARE Agreement