

# GUApp: Enhancing Job Recommendations with Knowledge Graphs

Discussion Paper

Giovanni Maria Biancofiore<sup>1,2</sup>, Tommaso Di Noia<sup>1</sup>, Eugenio Di Sciascio<sup>1</sup>,  
Fedelucio Narducci<sup>1</sup> and Paolo Pastore<sup>1,2</sup>

<sup>1</sup>*Polytechnic University of Bari, Bari, Italy*

<sup>2</sup>*Corresponding authors*

## Abstract

GUapp is a platform for job-postings search and recommendation for the Italian public administration. The platform offers recommendation services with the aim of matching user skills and requests with job positions available in a given period of time. GUapp implements a chatbot whose conversations are led by a domain-specific Knowledge Graph which allows users to interact with the app through natural language. Thanks to that, the search and recommendation process becomes incremental and the user can add new requirements at each stage of the interaction. In this paper we present GUapp, its recommender system, the chatbot, and the main ongoing work that will bring GUapp to become a Conversational Recommender System powered by a domain-specific Knowledge Graph.

## Keywords

Recommender System, Public Administration, Knowledge Graph, Conversational Recommender System

## 1. Introduction

The information overload is a well-known problem that impacts the digital experience of users when they need to find interesting items in a large set of possible options [1]. That is the case of looking for a book to read, a digital camera to buy, a movie to watch, etc. Another scenario when this problem is particularly felt is when a user is looking for a new job. In this context the only strategy people can adopt is to manually selecting interesting job calls, thus resulting in a clumsy user experience. In the literature, this problem has been investigated in the information-retrieval (IR) and recommender-system (RS) research areas. In the first case, systems do not take into account the user past preferences and retrieve the most relevant items according to the user query. In the second case a user profile is built, allowing to provide a *personalized* ranked list of items.

The GUapp platform has been designed to find and discover job positions among job offers in the Italian public administration<sup>1</sup>. The first version of GUapp's RS classified jobs according to their topics, obtained by means of the Latent Dirichlet Allocation (LDA) algorithm, and a k-Nearest Neighbor (k-NN) method was used to compute a personalized recommendation


---

*IIR 2021 – 11th Italian Information Retrieval Workshop, September 13–15, 2021, Bari, Italy*

✉ giovannimaria.biancofiore@poliba.it (G. M. Biancofiore); tommaso.dinoia@poliba.it (T. Di Noia);  
eugenio.disciascio@poliba.it (E. Di Sciascio); fedelucio.narducci@poliba.it (F. Narducci); paolo.pastore1@poliba.it  
(P. Pastore)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

<sup>1</sup>All the documents are freely available online at <https://www.gazzettaufficiale.it/30giorni/concorsi>

list [2]<sup>2</sup>. Moreover, GUapp offers a natural-language based interaction through a chatbot. The GUapp's chatbot allows the user to define her interests, describe her skills, and filter out results that did not match with her requirements. Although the first version of GUapp was undoubtedly effective from the recommendation perspective, there were a few limitations in terms of User Experience. The combination of k-NN and LDA, in fact, does not allow the system to explain to the user why an item was recommended. This because the latent features learned by the model are not easily interpretable. Hence, the dialogues were modeled as a filtering procedure for the recommendations list without interacting neither exchanging information with the RS itself. Users that converse with the system can only provide preferences from a limited set of features that are foreseen a priori, that are city and profession.

To beat these issues, we opted for a new Conversational Agent that converses with users exploiting a domain-specific Knowledge Graph (KG). The GUapp KG is obtained by merging some sub-graphs from state-of-the-art solutions like Dbpedia<sup>3</sup> and new triples scraped from external sources such as the ISTAT<sup>4</sup> website. On this line, conversations can reach a fine grained level of details regarding job aspects to recommend to users and enhance its expressiveness.

## 2. Related Work

The work proposed in this paper cuts across two principal research areas: recommender systems and conversational agents. Recommender Systems (RSs) can in turn be grouped into two main classes: Collaborative Filtering (CF) and Content-based (CB). To support the user during the decision making process, CF RS exploits the user community in order to identify interesting items for a given individual, following the intuition that similar users like similar items. Conversely, CB RSs are based on matching user preferences with items textual content [1].

GUapp implements a CB RS since its goal is to match the user requests/preferences with the textual description of the job call. The specific task of job recommendation is widely investigated in the literature [3]. Malinowski et al. [4] build a multi-slot profile with information about demographic data, job experience and IT skills. Similarly, in GUapp, this kind of information was first acquired from the LinkedIn social network. This strategy, useful for addressing the cold-start problem, is now implemented by means of conversations driven by a KG. Recommending items by means of a set of rules which verify whether the user tastes are satisfied or not was dealt at first by the Knowledge based RSs [5]. Here the RS performs reasoning on ontologies and Knowledge Bases (KBs) to find items that match the user needs. Carrer-Neto et al. [6] recommend movies that belong to the same ontological classes of the items that the user liked, and the final list is further processed by means of a social context derived from Social Networks in which the user is signed-up. Tarus et al. [7] exploit both ontological and CF features to compute the learning resources to recommend for some learner targets.

On the other hand, Conversational Agents (CAs) are software agents able to interact with the user through natural language. They can be classified in two main classes: end-to-end and modular systems [8]. The former are generally based on neural networks [9, 10, 11, 12] and they learn a dialog model from a set of past conversations [13]. The modular systems are pipeline-based agents composed of a set of modules, each with a specific function [14, 15, 16, 17].

---

<sup>2</sup>This paper is an extended abstract of [2]

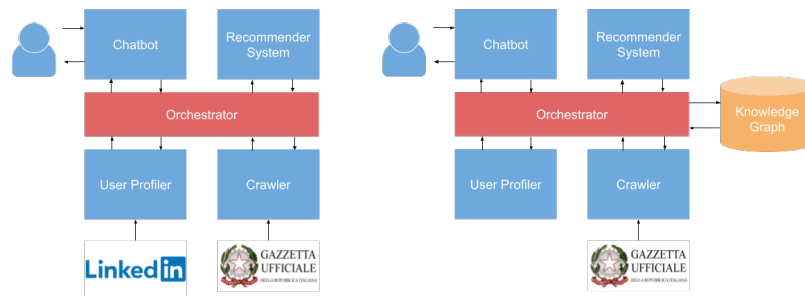
<sup>3</sup><https://www.dbpedia.org/>

<sup>4</sup><https://www.istat.it/>

The main difference between a Conversational Recommender System (CoRS) and a traditional RS is the interaction with the user that is more efficient and natural [18]. Accordingly, the construction of the user profile is incremental and the user can express her preferences at different stages of the recommendation process by an interactive human-like dialog [19]. The aforementioned task is well suited to Knowledge-based and KG-based RSs. An example can be found in [20], where a comprehensive KG is built upon a specific domain to lead dialogues and recommendations. Recently, CoRSs have been effectively used and tested in different domains such as music, movie, and book [21, 22]. However, to the best of our knowledge, GUapp represents the first attempt of implementing a CoRS for the job-recommendation task.

### 3. The GUapp's Architecture

In Figure 1 the initial GUapp's architecture is sketched together with the new one, with six main components: the Orchestrator, the Chatbot, the Recommender System, the User Profiler, the Crawler and the Knowledge Graph.



**Figure 1:** On the left, the first version GUapp's Architecture, on the right new one.

The *Orchestrator* manages the interaction between the different components of the system. For example, it invokes the recommender system when user asks for receiving a list of job positions based on the information stored in her profile.

The *Chatbot* is the component of GUapp which allows user to interact through natural language. It is powered by DialogFlow<sup>5</sup>, a Google platform for designing and integrating conversational user interfaces, and performs two main tasks: intent and entity recognition. The *Intent Recognizer* (IR) analyzes the request of the user expressed in natural language and understands her goal. E.g. if the user writes "I am looking for a new job in Bari" the IR identifies as intent *new\_job*. The GUapp's *Chatbot* adopts a mixed initiative interaction, allowing more fine grained conversations. For instance, the user may add her skills and professional titles through the *profile* intent. She may refine also her favourite job locations or occupations with the *refine\_location* and *refine\_occupation* intents respectively. The system can also take the initiative and ask for unspoken user features, that could eventually be inferred by the KG, and engage a negotiation about job postings that do not have a full match with the user requirements. In details, if the user profile is a tuple (*Bari, Computer Engineer, PhD*) and the retrieved results do not have a complete match, the system may ask: "What do you think about Computer Science in Bari?" or "Is it still good for you a position in Lecce?". Once the intent has been recognized, the *Entity Recognizer* (ER) is invoked. ER is implemented through DialogFlow as well, and it is

<sup>5</sup><https://dialogflow.com/>

powered by the KG entities. It adopts a fuzzy-matching strategy in order to identify real-world entities in the user sentence. In the case the match succeeds, the recognized entity is returned. In the example above, the entity is *Bari*, a city of southern Italy.

The *User Profiler* (UP) instead stores and updates information about user preferences and habits. Differences here with the first version of GUapp are on the information type stored in UP. We now collect data about all the user preferences provided during conversations instead of users' likes. Indeed, user can *refine* the retrieved list of relevant job positions as described above. Due to some variations on Microsoft's terms of use, the new version of GUapp does no more have access to the LinkedIn<sup>6</sup> data. At first we found useful to collect information stored in the social network to profile users by simply asking them a login with their LinkedIn credentials. This was an effective strategy for addressing the *cold-start* problem. Now, when the user signs in to the app for the first time, her profile is empty (i.e. *cold-start* situation). Thanks to the new chatbot powered by our KG, the user talks with GUapp about her skills, wishes and possibilities, then the system is able to rank job calls by exploiting the information that the user provided.

The *Recommender System* is another core component of GUapp. It exploits an Elasticsearch<sup>7</sup> index, which stores all the information scraped by the *Crawler* enriched with all the linked entities of our KG. We are now stressing the choice of a similarity function that effectively returns a ranked list of item based on the data collected by the UP. The main intuition behind adopting this model is the possibility to make dialogues as interactive and efficient as possible, allowing the system to negotiate with users about results that do not completely match with their preferences. The UP is actively updated to the new user inputs, guaranteeing the recommendations to be adaptive. As a matter of fact, GUapp implements a *Crawler* that daily extracts the job positions from *Gazzetta Ufficiale*, the official journal of record of the Italian government.

Finally, the system has been equipped with a domain-specific *Knowledge Graph* built upon several sources, like Dbpedia and the ISTAT website as stated before. The latter identifies a collection of raw data that are not in form of KG. This leads us to model a new *Ontology* on which the overall KG can rely. The GUapp *Ontology* in fact defines relations and hierarchies about professions, domain of competences, locations and so forth wisely integrating the raw information with the already structured ones. To make the system more specialized on the job-opening domain, we plan to enrich our KG of further facts related to the job recommendation task. This KG allows the system to have natural dialogues with users about the discussed domain and the quality of recommendations results improved as well.

## 4. Conclusion and Future Work

In this paper we presented a platform for searching jobs in the Italian public administration and the main ongoing work for improving its performances and the user experience. GUapp consists of a recommender system that suggests relevant jobs to the users. One of the most interesting aspects of the new version is the integration of a KG that helps driving the dialog, making the interaction more natural. Indeed, the preference elicitation becomes incremental, with the possibility of refining and improving the user requests. We are planning of introducing several fine-grained features for making the ecosystem competitive with other state-of-the-art solutions, as well testing the User Experience with an A/B test.

---

<sup>6</sup><https://www.linkedin.com/>

<sup>7</sup><https://www.elasticsearch.com>

## References

- [1] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, *Recommender systems: an introduction*, Cambridge University Press, 2010.
- [2] V. Bellini, G. M. Biancofiore, T. D. Noia, E. D. Sciascio, F. Narducci, C. Pomo, Guapp: A conversational agent for job recommendation for the italian public administration, in: *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2020, Bari, Italy, May 27-29, 2020*, 2020, pp. 1–7. URL: <https://doi.org/10.1109/EAIS48028.2020.9122756>. doi:10.1109/EAIS48028.2020.9122756.
- [3] S. T. Al-Otaibi, M. Ykhlef, A survey of job recommender systems, *International Journal of the Physical Sciences* 7 (2012) 5127–5142.
- [4] J. Malinowski, T. Keim, O. Wendt, T. Weitzel, Matching people and jobs: A bilateral recommendation approach, in: *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 6, IEEE, 2006, pp. 137c–137c.
- [5] R. Burke, Knowledge-based recommender systems, *Encyclopedia of library and information systems* 69 (2000) 175–186.
- [6] W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García, F. García-Sánchez, Social knowledge-based recommender system. application to the movies domain, *Expert Syst. Appl.* 39 (2012) 10990–11000. URL: <https://doi.org/10.1016/j.eswa.2012.03.025>. doi:10.1016/j.eswa.2012.03.025.
- [7] J. K. Tarus, Z. Niu, A. Yousif, A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining, *Future Gener. Comput. Syst.* 72 (2017) 37–48. URL: <https://doi.org/10.1016/j.future.2017.02.049>. doi:10.1016/j.future.2017.02.049.
- [8] A. Bakarov, V. Yadrintsev, I. Sochenkov, Anomaly detection for short texts: Identifying whether your chatbot should switch from goal-oriented conversation to chit-chatting, in: *International Conference on Digital Transformation and Global Society*, Springer, 2018, pp. 289–298.
- [9] L. Shang, Z. Lu, H. Li, Neural responding machine for short-text conversation, *arXiv preprint arXiv:1503.02364* (2015).
- [10] O. Vinyals, Q. Le, A neural conversational model, *arXiv preprint arXiv:1506.05869* (2015).
- [11] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, B. Dolan, A neural network approach to context-sensitive generation of conversational responses, *arXiv preprint arXiv:1506.06714* (2015).
- [12] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, J. Weston, Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems, *arXiv:1511.06931 [cs]* (2015). URL: <http://arxiv.org/abs/1511.06931>.
- [13] A. Bordes, Y.-L. Boureau, J. Weston, Learning end-to-end goal-oriented dialog, *arXiv preprint arXiv:1605.07683* (2016).
- [14] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, J. Weston, Evaluating prerequisite qualities for learning end-to-end dialog systems, *arXiv preprint arXiv:1511.06931* (2015).
- [15] T. Zhao, M. Eskenazi, Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning, *arXiv:1606.02560 [cs]* (2016). URL:

<http://arxiv.org/abs/1606.02560>, arXiv: 1606.02560.

- [16] J. Williams, A. Raux, M. Henderson, The dialog state tracking challenge series: A review, *Dialogue & Discourse* 7 (2016) 4–33.
- [17] B. Liu, I. Lane, An End-to-End Trainable Neural Network Model with Belief Tracking for Task-Oriented Dialog, *Interspeech 2017* (2017) 2506–2510. URL: <http://arxiv.org/abs/1708.05956>. doi:10.21437/Interspeech.2017-1326, arXiv: 1708.05956.
- [18] P. Wärnestål, User evaluation of a conversational recommender system, in: *Proceedings of the 4th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2005, pp. 32–39.
- [19] M. Jugovac, D. Jannach, Interacting with recommenders: Overview and research directions, *ACM Trans. Interact. Intell. Syst.* 7 (2017) 10:1–10:46.
- [20] S. Hausmann, O. Seneviratne, Y. Chen, Y. Ne’eman, J. Codella, C. Chen, D. L. McGuinness, M. J. Zaki, Foodkg: A semantics-driven knowledge graph for food recommendation, in: *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II, 2019*, pp. 146–162. URL: [https://doi.org/10.1007/978-3-030-30796-7\\_10](https://doi.org/10.1007/978-3-030-30796-7_10). doi:10.1007/978-3-030-30796-7\_10.
- [21] F. Narducci, P. Basile, M. de Gemmis, P. Lops, G. Semeraro, An investigation on the user interaction modes of conversational recommender systems for the music domain, *User Modeling and User-Adapted Interaction* (2019) 1–34. doi:<https://doi.org/10.1007/s11257-019-09250-7>.
- [22] A. Iovine, F. Narducci, G. Semeraro, Conversational recommender systems and natural language: A study through the converse framework, *Decision Support Systems* (2020) 113250. doi:<https://doi.org/10.1016/j.dss.2020.113250>.