IRnator: A Framework for Discovering Users Needs from Sets of Suggestions

Agnès Mustar
Sorbonne Université, ISIR
Paris, France
mustar@isir.upmc.fr

Sylvain Lamprier
Sorbonne Université, ISIR
Paris, France
sylvain.lamprier@isir.upmc.fr

Benjamin Piwowarski
Sorbonne Université, ISIR, CNRS
Paris, France
benjamin.piwowarski@isir.upmc.fr

ABSTRACT

To tackle complex IR tasks, where users cannot precisely define their needs, interaction is paramount. Both query-reformulation approaches and chatbots are limited for this type of task, since the former only learn to mimic users, while the latter are bounded by the domain they have been trained on. To take a first step towards truly exploratory and interactive IR, we introduce a framework, where users navigate document collections by expressing their preference among sets of queries proposed by the system at each step – thus refining the knowledge about the user's information need. Our training approach, based on self-supervised and reinforcement learning techniques, aims at minimizing the amount of interactions required to reach relevant queries, and thus documents, for users. We experimentally show that the introduced framework enables efficient learning from interactions with simple user bots, that are demonstrated to generalize well in real-world settings.

CCS CONCEPTS

• **Information systems** → **Query intent**; *Query suggestion*; Search interfaces; Information retrieval diversity; Clustering and classification.

KEYWORDS

Interactive Search, Query Intent, Information Retrieval Framework, User Need, User Behavior

ACM Reference Format:
Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. 2022. IRnator: A Framework for Discovering Users Needs from Sets of Suggestions. In *Proceedings of the 2022 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '22), July 11–12, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3539813.3545152

1 INTRODUCTION

For complex search tasks, when user needs cannot be precisely defined from a single query, interaction with session-based Information Retrieval systems is essential. Different session-based IR models have been proposed [17, 18, 31], but they focus on biasing the document ranking process, thus preventing the user to truly interact with the system. More direct interactions can be provided using query suggestions approaches [9, 20, 25], that help users by reformulating their needs from interactions during the session. Most of them are based on behavior models to predict the next queries of search sessions. Finally, chatbots for Information Retrieval, while ambitious in their goals, are usually adhoc systems, that are restricted to simple dialogues for the specific domain they have been trained for [8].

Going further supposes IR systems able to *anticipate* user behavior so that they can pro-actively help users in their search tasks, as well as systems that can consider various possibilities in the evolution of the search process. To take a first step in this direction, we study in this paper a simpler problem, inspired by Akinator-like systems [13], i.e. systems that find a user's intent by asking questions about it. More specifically, we propose a system that interacts with the user by proposing queries among which users choose the one that reflects the best their need (see Figure 1). We argue that this type of task requires an IR system that refines its knowledge about users' needs to guide them more quickly towards relevant queries. In time, such systems could be turned into powerful conversational agents for IR.

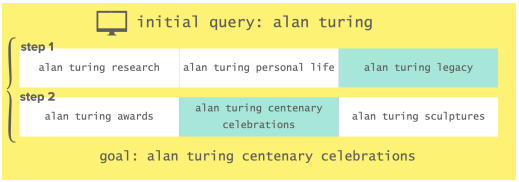

Figure 1: IRnator: the proposed framework

Our method differs from query suggestion in two ways. First, query suggestion focuses on one or a few steps [9, 12, 20, 25, 29] of a search session. In contrast, we aim at helping users to fulfill their information needs. Second, while query suggestion works mostly focus on behavioral cloning methods, wherein the agent learns to mimic the user by predicting future actions, we aim at explicitly shortening the user efforts. We argue that this is necessary since users do not necessarily know the best course of actions to reach relevant documents. Finally, the data needed to train query suggestion models are based on search session logs. These logs are expensive to obtain, and raise serious questions about user privacy. They are dependent on the search engine used by the user at the time of extraction, and do not allow the model to generalize if new goals or queries arise. It is thus interesting to develop models that do not rely on this type of data.

In this paper, after the introduction of this navigation framework, we propose a training approach, based on self-supervised and reinforcement learning techniques, that attempts to minimize

ICTIR '22, July 11–12, 2022, Madrid, Spain

ACM ISBN 978-1-4503-9412-3/22/07…$15.00
https://doi.org/10.1145/3539813.3545152

amounts of interactions required to reach relevant queries for users. Our research question is whether it is possible to train an IIR system with simulated user that generalizes to real-world setting. Our experiments show promising results.

2 RELATED WORKS

Our work is at the crossroads of many: asking clarifying questions, query suggestion, interactive IR systems, and user simulation.

Several works have proposed to ask questions about users' goals to infer them. In particular, [5, 14, 29] study the Q20 game, and [33] the Akinator game, where the agent asks questions about the goal. In our setting, the main difference is that the search space is much higher, and there are no predefined attributes that can guide the search. More IR-related, [10] disambiguate an initial query by asking a question to discriminate the most likely intent, but in a one-step interactive process that can only be applied when the number of intents is small.

Some tasks such as product recommendation [4, 32] bear some similarities with our work, since they aim at predicting user intents. These works are generally based on item (category, etc.) and user metadata (gender, age, location, etc.) and interaction logs. Our research direction is orthogonal since we focus on a session-based single intent prediction – beside not using any metadata and/or interaction log.

Query suggestion works [9, 12, 20, 25, 29] model users' sessions so as to predict their next query, which is then used as a suggestion. Most of these works do not take into account user feedback, except [29] who use clicked (or not) documents. These works consider that the user's future queries are relevant suggestions. In contrast, we view query suggestions as a way to uncover the user intent.

Closer to our work, interactive search sessions have already been modeled [17, 18, 31] as a MDP (Markov Decision Process), in which the search engine plays the role of the agent. These works focus on ranking documents, and not on the interaction with the user, which could provide a better understanding of the user's goal. For instance, [31] studies the user behaviour by focusing on the syntactic query changes during a session and doesn't provide the user additional information. While [22] uses a setting closer to ours, it learns a strategy to reach the user's goal as quickly as possible. However, it works with structured data (with a hierarchy), and requires conversational data.

Finally, [1, 3, 6, 19, 27] attempt to simulate users, based on a more or less complete description of the user's need. While simulating IR users in an interactive setting is a crucial topic to develop better interactive IR systems, such models are still difficult to use and not so reliable. In this work, we rely on a simple user heuristic, that allows to get a large number of simulated sessions needed for training our model and leave for future work the use of more sophisticated models.

3 A PREFERENCE-BASED IR FRAMEWORK

Rather than directly attempting to answer the user need, which is usually ill-defined for complex needs, or trying to have a conversation with the user about its interests, which is very difficult to efficiently drive and interpret, we introduce a new kind of interaction methodology, where the IR system successively proposes K query suggestions among which users can choose the one that best reflects their need. We think that this task, while simple, if successfully conducted, can be the basis of more ambitious conversation-based IR models because it (1) supposes we can uncover the user intent; and (2) requires that the system proposes different paths the user can follow.

Formally, let us consider a session S composed of $|S|$ interaction steps between a user ξ with a goal g and an IR system π. We suppose that the session starts with an initial query q_0, which follows a distribution $\xi_0(g)$ of initial queries for the user ξ having a need g. Each interaction step t corresponds to $S_t = (Q_t, u_t)$, where $Q_t = \{q_t^1, .., q_t^K\}$ corresponds to a set of K query suggestions, and u_t is the index of the user's preferred suggestion amongst the K. A complete session is denoted as $S = (q_0, S_1, \ldots, S_{|S|})$.

Any user choice u_t of a given session follows a conditional distribution about preferences of the user given the goal and the session up to step t, i.e. $u_t \sim \xi(u_t|g, S_{<t}, Q_t)$, where $S_{<t}$ denotes all interactions before step t in the session. Successive sets of questions suggested by the system also follow a conditional distribution $\pi(Q_t|S_{<t})$ given the past of session $S_{<t}$ at step t. Finally, a session S with a goal g follows a distribution $\mathcal{S}_\pi^\xi(g)$, depending both on the user model ξ and the suggestion system π.

The aim is to suggest query sets Q_t that allow to increase information about g as much as possible at each step, to help users achieving their goal as soon as possible. We introduce an interactive IR system whose aim is defined as the following optimization, given sessions with a maximum number of interactions T:

$$\pi^* = \arg\max_\pi \mathop{\mathbb{E}}_{g\sim\mathcal{G}} \mathop{\mathbb{E}}_{S\sim\mathcal{S}_\pi^\xi(g)} \left[\sum_{t=0}^{T} \gamma^t Achieved(g, S_{\leq t})\right] \quad (1)$$

where $\mathcal{G}$ is the distribution of goals and $Achieved(g, S_{\leq t})$ is a binary function that returns 1 if goal g can be directly completed given information from $S_{\leq t}$, and 0 otherwise, depending on the considered IR system. The proposed framework can consider complex goals, which implies for instance the interrogation of a document retrieval system given the last selected query (or the full past session) and the inspection of the corresponding returned documents to assess completion of g. In Eq.1, $\gamma \in]0; 1[$ is a discount factor that pushes to prefer sessions which complete goal g as soon as possible. In this paper, for the sake of simplicity, and to avoid the dependence on a document collection with its specific retrieval system, we consider that goal g can be expressed as a query q_g and that g is achieved at step t if the system π proposes a set of suggestions that includes q_g i.e., $q_g \in \{q_t^1, .., q_t^K\}$.

The problem as defined in Eq.1 is however particularly difficult to directly solve using standard Reinforcement Learning algorithms, as it involves the following challenges:

Query space size. Ideally, given a vocabulary V of $|V|$ tokens and a max query length L, any suggested query q lives in V^L. This is huge, even for reasonably-sized vocabularies, and includes many sequences that do not correspond to human-readable queries (e.g., with token sequences that form words that do not exist in the user's language). While a prior query-language model could be used, in this work we simplify the task for π, by only restraining suggested queries to a set $\boldsymbol{Q}$ of pre-defined ones, from which the system samples sub-sets at each step, which allows to greatly restrict the

search space. Dealing with more complex (generative) strategies is left for future works.

Combinatorial action space. Even with the reduction of the query space as proposed above, the action space remains particularly large, because of its combinatorial aspect: an action for π corresponds to select K queries from Q, inducing an action space of size Q^K. While a policy π composed of a main network (e.g., Transformer) with K heads on top of its output would be an option, this still implies a complex search space, involving a hard credit assignment problem, well known in the multi-agent RL literature [11]. As detailed in the following section, we assume a well structured semantic representation space of queries, that reduces the choice of Q_t to a single point in the space, from which the set of K suggestions can be deterministically determined (here, by clustering queries).

User model ξ unknown. Modeling users of interactive IR systems is a particularly difficult task [7]. Beyond the lack of training IR session data, especially when considering innovative systems like the one we introduce in this paper, behaviors of users are very difficult to precisely predict in many settings, due to the implication of many confounding factors. While it is well known that behaviors are not stationary during IR sessions, we assume here that past interactions do not modify user's preferences during the search. Moreover, rather than modelling complex user behaviors, as done for instance in classical – short term – query suggestion [20], we assume in the following a simple user bot as ξ, hard-coded with pre-defined heuristics shared across sessions, though possibly hidden from the system agent π to be general enough for application of the model in real-world settings (where minds of users are not accessible).

Very sparse reward problem. As defined in Eq.1, system π must succeed in generating a target query in less than T steps to expect a non-null reward. Thus, in first steps of learning, no improvement direction of π are given to the learner, preventing it from completing the task. Reward shaping [21] is a popular way to densify rewards for such hard problems, where advisories about states to visit are given as potential functions $\phi : \mathcal{S} \rightarrow \mathbb{R}$, with $\mathcal{S}$ the set of reachable states in the environment[1]. In addition to a self-supervised learning process to initiate the learning process, we consider in the following a learnt model of user intent prediction ϕ_g, based on the partial user sessions, to drive the learning of π following directions which minimize the uncertainty of g with respect to this model.

Note that, assuming a well-known user that deterministically selects the closest suggestion to its goal in its own euclidean representation space ψ^{ξ}, the problem as defined in Eq.1 could be greedily optimized by choosing each step t the set of queries that minimizes the number of admissible goals regarding $S_{\leq t}$. Also, for a probabilistic user, the optimal solution could be approximated by suggesting at each step t the set of queries $Q_t = \{q_t^1, .., q_t^K\}$ that minimizes the conditional entropy $H(G|U_t) = \sum_{u=1}^{K} \xi(u|Q_t, S_{<t}) H(G|S_{\leq t})$, with $\xi(u|Q_t, S_{<t})$ the marginal probability that the user selects the query of index u given Q_t and the past of session $S_{<t}$, and $H(G|S_{\leq t}) = -\sum_g \phi(g|S_{\leq t}) \log \phi(g|S_{\leq t})$ the entropy of goal distribution given session $S_{\leq t}$. However, while this can be considered for instance for interactive classification with restricted sets of labels and closed questions, such as in [33], this is completely intractable in our setting.

[1] In our setting, $\mathcal{S}$ corresponds to the full set of possible search sessions that can be built for any user from the set of all possible needs.

4 LEARNING TO DRIVE USERS TOWARDS GOALS

This section first presents the considered suggestion architecture π, before describing self-supervised and reinforcement learning techniques used to solve the task.

4.1 Query suggestion process

Let us consider that the set of all possible queries $q \in Q$ belong to a continuous representation space, i.e. $\psi(q) \in \mathbb{R}^d$. Figure 2 depicts the proposed suggestion process, where π is implemented as a Transformer architecture [28], which takes as input the session S as input and outputs a set of K suggestions (in the figure, $K = 3$). To provide a diverse set of suggestions, we rely on a clustering process based on a point $\pi(S)$ predicted by our model. The N closest queries from Q (queries are represented by crosses in the figure), depending on euclidean distances in the continuous space ψ, are selected and clustered into K groups. Finally, the K medoids of clusters are used as the set of queries Q_t proposed to the user at step t. The user selects their preferred query, depending on g and ξ (q_3^3 in our example, which is the closest suggestions to g). This feedback u_t defines S_t that is used for the next suggestion step.

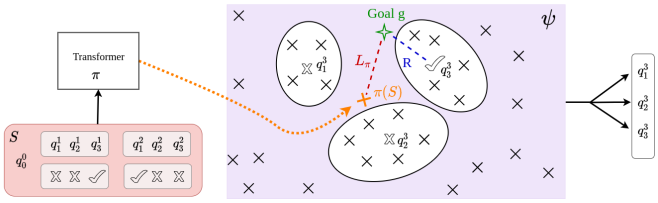

Figure 2: Query suggestion process

The assumption behind the use of a clustering method (a simple K-means approach in our experiments) is that the neighborhood $\pi(S)$ in ψ contains the main aspects that can specialize $\pi(S)$, which can be partitioned in relevant sub-topics to present to the user. We argue that, while the use of hyperbolic representation spaces [26] could allow to even improve accuracy (which we leave for future works), the representation space ϕ we consider, which results from a pre-trained sentence-transformer designed for semantic search [24], presents a structure that fits well with this assumption, with general queries tending to occupy central positions in the representation space.

The suggestion model π corresponds to a Transformer architecture [28], which takes as input sessions concatenation of the initial query with all past interactions, each S_t being encoded as the sum of three representations: query embeddings, coming from a FAISS index [15] on top of query representation ϕ, positional embeddings, allowing to retain the temporality of interactions, and the user's action embeddings, corresponding to the user's choice (1

for selected queries, 0 for the others). This model π is trained using two learning modes, that we describe in next sections.

4.2 Iterative Supervision

The first mode considered for training our model is an iterative supervised learning, where we seek at minimizing the euclidean distance between the point $\pi(S)$ predicted by the model and the user's final goal (represented as L_π on figure 2), given various input pairs of (goal, session) as input. At each iteration i of the training algorithm, the following optimization problem is considered:

$$\arg\min_{\pi} \sum_{(g,S)\in\Gamma^{(i)}} ||\psi(g) - \pi(S)||_2^2 \quad (2)$$

where $\psi(g)$ returns the representation of the query targeted by goal g (for simplicity, we focus here on the case where a goal corresponds to a single target query), and $\Gamma^{(i)}$ is the training set at iteration i, obtained using the distribution of goals $\mathcal{G}$ and the policy $\pi^{(i-1)}$, obtained at iteration $i-1$ of the learning, π^0 being a random suggestion policy. At step i, after optimization of Eq. 2, ψ is used as the new policy π_{i+1}.

4.3 Reinforcement Learning

While the iterative supervised learning proposed in previous section enables to train the model accurately, this may suffer from different limitations: 1) no convergence guarantee due the iterative process which does not take into account the dependence of the training data on the optimized model; 2) strong relatedness with the user heuristics, which prevents from the ability to adapt to different kinds of users; 3) no direct consideration of the queries presented to the user.

Thus, we propose here to consider possible refinement of the supervised model via reinforcement learning techniques, notably DDPG [2], a policy gradient approach specifically designed for continuous actions as it is the case for our setting where the action corresponds to outputting point $\pi(S)$. As previously mentioned, to deal with sparse rewards, and to gain in flexibility regarding the considered user, we propose to consider a probabilistic intent model $\phi(g|S_{\leq t})$ as the intrinsic reward at each step t, implemented as a Transformer that outputs the mean vector μ of a Gaussian $\mathcal{N}(\mu, I)$ with unit variance. This allows to reward suggestion sets that most improve knowledge about the hidden user's goal, according to the user's answer: $R_t = \log\phi(g|S_{\leq t}) - \log\phi(g|S_{<t})$, where the second term acts as a baseline. The intent model is refined for some iterations at each optimization epoch, to update it regarding distributions of sessions, via goal likelihood maximization.

Finally, rather than dealing with long term reinforcement, which appeared unstable in our experiments, we propose to use a one-step ahead critic network $Q(S_{<t}, \pi(S_{<t}))$, that simply learns to predict R_t from past interactions and the output of the suggester π.

5 EXPERIMENTS

5.1 Experimental Details

Data. For our experiments, we use the Wikipedia dump from TREC CAR 2020 [23], which is interesting because it covers a large spectrum of domains. Wikipedia page titles are used as initial queries, and the names of the sub-sections are concatenated to the title to obtain final goals. For example, for the page `'anarchism'` which contains a section `'history'` with a sub-section `'prehistoric and ancient world'`, we get an initial query-goal pair: (`'anarchism'`, `'anarchism prehistoric and ancient world'`). This method provides general initial queries and specific goals. As the latter are specific, they would probably not have been the initial query of a user. An important advantage of this method is that an initial query can lead to different goals, so the model *must* actually learn to suggest discriminating queries and to use the user's answers (rather than relying only on the first query). Furthermore, to focus on rather complex search goals, we only kept pages with at least 3 sections, each containing at least 2 sub-sections. Sections with too long titles (more than 3 words) or too generic – e.g. `'see also'`, `'references'`, `'citations'`, `'sources'`, `'further reading'`, `'external links'`, `'notes'`, `'other'`, `'notes and references'` – are also filtered out. Following this process, we get 633,647 pairs that we split into train and test with a 80-20 ratio. The data is split so that there is no common goal between the train set and the test set: test goals were never seen during the training phase The scores reported are computed on the test set. Data will be released upon acceptance.

Note that the database can be easily expanded with other queries from different sources. We are aware that using synthetic data has its shortcomings, but using (filtered) query logs would have introduced too much noise, preventing analyzing the model behavior in such a controlled setting.

	SC	*SS*	SSC_{rand}	*SSC*	*DDPG*
% success	0.175	0.178	0.308	0.457	**0.475**
# steps	5.212	5.215	5.080	4.571	**4.568**
min. dist	0.371	0.409	0.415	0.259	**0.253**

Table 1: Models scores

Compared Models. In our experiments, unless specified otherwise, we use a simple model to simulate the user's choices, both at train and test time (except for the human evaluation experiment): at each step, our bot user chooses the closest query (in term of euclidean distance) to its target goal in the representation space ψ. We compare our Self-Supervised Suggestion model with Clustering (*SSC*) with three of its ablations. We remove the suggestion model π in the first ablation (*SC*), and use the previous user's choice to obtain next suggestions, i.e. $\pi_{SC}(S_{<t}) = \psi(q_{t-1}^{u_{t-1}})$. The second ablation (*SS*) removes the clustering step, and replaces it by proposing the K queries closest to $\pi(S_{<t})$. Finally, the last ablation (SSC_{rand}) considers a random user for supervision rather than our heuristic bot user described above. Finally, we also consider a policy fine-tuned via RL (*DDPG*), as described in section 4.3.

All models use $K = 3$ and a maximal session length of $T = 6$. The policy and the intent models all have the same architecture: a Transformer with 6 heads and 6 layers and a dropout $p = 0.1$. We use a feedforward network with two layers with hidden size of 768, which corresponds to the size of the embeddings in the FAISS index, to compute $\pi(S_{<t})$ from the contextualized `CLS` token. The model is optimized with Adam [16] – we observed that the Self-Supervised model converged quickly after a few steps (around 5-10).

5.2 Results

Suggestion policies are evaluated in terms of average success (i.e., the rate of sessions where the target query was finally suggested by the system within the T=6 steps of interaction), average number of steps to complete the task (using 6 if the goal was not reached) and minimum distance (i.e., the average distance between the closest suggestions and the target in each session).

Performances of Compared Strategies. Table 1 reports results of the compared policies π. First, the *SC* ablation obtains the worst results, which indicates that simply focusing on the neighborhood of expressed or selected user queries is not enough to help navigation, validating the usefulness of the learning task. Second, the *SS* ablation does not demonstrate significantly better results, which points out the relevance of the use of a clustering to ensure diversity of suggestions. Third, and very importantly, the SSC_{rand} approach obtains significantly worse results than our *SSC*, which shows that the latter succeeds in leveraging useful feedbacks of users, only suggesting using the initial query and the structure of available ones is not enough. Finally, the reinforcement learning approach *DDPG* allows us to obtain the best results, with no big improvements over the self-supervised approach *SSC*, but showing the potential of using such a way more flexible learning paradigm that RL enables.

Human Evaluation. To analyze if models trained with our heuristic user can be helpful for real users interacting with the system, we asked three annotators to use IRnator. At the beginning of each session, they are given an initial query and a goal to reach (see section 5.1). At each step, they are asked to select the proposition that corresponds the best to their final goal. Their aim is to navigate towards the specified target, only via selecting at each step a query among the three proposed. Suggestions are randomly proposed by one of the compared model, hidden from the annotator. The results are presented in table 2 with 150 samples per model.

While the variance is high, we see that the general magnitude of the measures corresponds to the model scores with the simulated user, confirming the validity of our approach. The only difference is for *SS* (no clustering) – which can be explained because it is much harder for a human to know which query is closer to the target when they are not enough diverse, and for DDPG, which shows that we need more realistic user models to generalize better.

The only difference is for *SS* (no clustering) – which can be explained because it is much harder for a human to know which query is closer to the target when they are not enough diverse, and for DDPG, which shows that we need more realistic user models to generalize better. We further discuss these points in Section 6.

	SS	*SC*	*SSC*	*DDPG*
% success	0.06	$0.21^{\star}$	$0.5^{\star\dagger}$	$0.38^{\star\dagger}$
# steps	5.71	$5.17^{\star}$	$4.25^{\star\dagger}$	$4.83^{\star}$
min. dist	0.48	$0.34^{\star}$	$0.23^{\star\dagger}$	$0.32^{\star}$

Table 2: Human evaluation. $\star$ indicates significant gains ($p < 0.05$) compared to *SS*. $\dagger$ indicates significant gains ($p < 0.05$) compared to *SC*.

6 DISCUSSION

IRnator is a generic framework for interactive search, which allows to study how an agent can guide a user in a knowledge space so that they reach their goal with minimal effort. We believe that, for a search engine, the challenge of learning to interact with a user is ambitious and requires simplifications that we restate and justify below.

User model. Our user behavior is stationary (it does not depend on the previous interactions) and relies on heuristics. These simulated users are always able to choose the query closest to their goals (in the representation space). In reality, it might happen that none of the proposed queries matches what the user wants or that the user does not know which query is the best. We should study the possibility for the user to submit a new query, or to express a negative feedback on the suggestions, rather than being forced to choose a proposition. Future works should explore more realistic user models, with more possible actions. However, even with such a simplified setting, we show in our human evaluation experiment that there exists a correlation between real and simulated users in terms of reduction of the effort to reach the goal.

Discrete query space. We use a space with a finite number of queries to focus on the agent role as a guide towards the goal rather than dealing with text generation problems. However, in our experiments the database contains a large number of query/goals (633,647) from a Wikipedia dump, a website that covers many domains. The scores presented are from the test set, thus based on goals never seen in the training phase. This shows the generalization capacity of our model: the agent has learned to navigate in this knowledge space. The large size of the chosen space and the ability to generalize to new goals, allow us to think that simplifying the space to a finite number of queries is acceptable.

7 CONCLUSION

We introduced the IRnator framework, inspired from Akinator systems [30], for the context of complex search sessions in information retrieval. The aim of the system is to guess the hidden user's intent by suggesting sets of query suggestions and leveraging its feedbacks. Rather than hard-coding non-scalable suggestion heuristics, based for instance on conditional entropy minimization, the associated learning task aims at discovering efficient strategies according to the user's behavior. An efficient clustering-based solution on top of a Transformer architecture, learned via self-supervised and reinforcement learning, was proposed as a first solution for this innovative task. We expect many promising directions for this very challenging, but crucial, problem of intent discovery in IR.

ACKNOWLEDGMENTS

This work was supported by the Agence National de la Recherche (ANR), through project CoST, code ANR-18-CE23-0016.

REFERENCES

[1] Leif Azzopardi. 2014. Modelling interaction with economic models of search.

[2] Gabriel Barth-Maron, Matthew W. Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva TB, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. 2018. Distributional Policy Gradients. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SyZipzbCb

[3] Feza Baskaya, Heikki Keskustalo, and Kalervo Järvelin. 2013. Modeling behavioral factors ininteractive information retrieval.

[4] B. Bhattacharya, I. Burhanuddin, A. Sancheti, and K. Satya. 2017. Intent-Aware Contextual Recommendation System. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE Computer Society, Los Alamitos, CA, USA, 1–8. https://doi.org/10.1109/ICDMW.2017.8

[5] R Burgener. 2006. 20q: The neural network mind reader. In *Goddard Space Flight Center Engineering Colloquium*.

[6] Arthur Câmara, David Maxwell, and Claudia Hauff. 2022. Searching, Learning, and Subtopic Ordering: A Simulation-based Analysis. *CoRR* abs/2201.11181 (2022). arXiv:2201.11181 https://arxiv.org/abs/2201.11181

[7] Arthur Câmara, David Maxwell, and Claudia Hauff. 2022. Searching, Learning, and Subtopic Ordering: A Simulation-based Analysis. *arXiv preprint arXiv:2201.11181* (2022).

[8] Cen Chen, Chilin Fu, Xu Hu, Xiaolu Zhang, Jun Zhou, Xiaolong Li, and Forrest Sheng Bao. 2019. Reinforcement Learning for User Intent Prediction in Customer Service Bots. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1265–1268. https://doi.org/10.1145/3331184.3331370

[9] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *Proceedings of the 26th ACM International on Conference on Information and Knowledge Management (CIKM '17)*. ACM, New York, NY, USA, 1747–1756.

[10] Kaustubh D. Dhole. 2020. Resolving Intent Ambiguities by Retrieving Discriminative Clarifying Questions. arXiv:2008.07559 [cs.AI]

[11] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2017. Counterfactual Multi-Agent Policy Gradients. *arXiv:1705.08926 [cs]* (May 2017). http://arxiv.org/abs/1705.08926 arXiv: 1705.08926.

[12] Vikas K. Garg, Inderjit S. Dhillon, and Hsiang-Fu Yu. 2019. Multiresolution Transformer Networks: Recurrence is Not Essential for Modeling Hierarchical Structure. (2019). arXiv:1908.10408

[13] Adrian Groza and Loredana Coroama. 2019. A mentalist agent for identifying characters using dynamic query strategies. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 319–326.

[14] Huang Hu, Xianchao Wu, Bingfeng Luo, Chongyang Tao, Can Xu, Wei Wu, and Zhan Chen. 2018. Playing 20 question game with policy-based reinforcement learning. *arXiv preprint arXiv:1808.07645* (2018).

[15] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

[17] Jiyun Luo, Xuchu Dong, and Hui Yang. 2015. Session Search by Direct Policy Learning. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval (ICTIR '15)*. Association for Computing Machinery, New York, NY, USA, 261–270. https://doi.org/10.1145/2808194.2809461

[18] Jiyun Luo, Sicong Zhang, Xuchu Dong, and Hui Yang. 2015. Designing States, Actions, and Rewards for Using POMDP in Session Search. In *Advances in Information Retrieval (Lecture Notes in Computer Science)*, Allan Hanbury, Gabriella Kazai, Andreas Rauber, and Norbert Fuhr (Eds.). Springer International Publishing, Cham, 526–537. https://doi.org/10.1007/978-3-319-16354-3_58

[19] David Maxwell and Leif Azzopardi. 2016. Simulating Interactive Information Retrieval: SimIIR: A Framework for the Simulation of Interaction. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Pisa, Italy) *(SIGIR '16)*. Association for Computing Machinery, New York, NY, USA, 1141–1144. https://doi.org/10.1145/2911451.2911469

[20] Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. 2021. On the study of transformers for query suggestion. *ACM Transactions on Information Systems (TOIS)* 40, 1 (2021), 1–27.

[21] Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. 1999. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML '99)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 278–287. http://dl.acm.org/citation.cfm?id=645528.657613

[22] Vishal Pallagani and Biplav Srivastava. 2021. A Generic Dialog Agent for Information Retrieval Based on Automated Planning Within a Reinforcement Learning Platform. *Bridging the Gap Between AI Planning and Reinforcement Learning (PRL)* (2021).

[23] Jordan Ramsdell and Laura Dietz. 2020. A Large Test Collection for Entity Aspect Linking. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 3109–3116.

[24] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[25] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*. ACM, New York, NY, USA, 553–562.

[26] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Hyperbolic representation learning for fast and efficient neural question answering. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 583–591.

[27] Paul Thomas, Alistair Moffat, Peter Bailey, and Falk Scholer. 2014. Modeling decision points in user search behavior. In *Proceedings of the 5th Information Interaction in Context Symposium (IIiX '14)*. Association for Computing Machinery, New York, NY, USA, 239–242. https://doi.org/10.1145/2637002.2637032

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[29] Xianchao Wu, Huang Hu, Momo Klyen, Kyohei Tomita, and Zhan Chen. 2018. Q20: Rinna riddles your mind by asking 20 questions. *Japan NLP* (2018).

[30] Qing Xie, Feng Xiong, Tian Han, Yongjian Liu, Lin Li, and Zhifeng Bao. 2018. Interactive resource recommendation algorithm based on tag information. *World Wide Web* 21, 6 (2018), 1655–1673.

[31] Hui Yang, Dongyi Guan, and Sicong Zhang. 2015. The Query Change Model: Modeling Session Search as a Markov Decision Process. *ACM Trans. Inf. Syst.* 33, 4, Article 20 (may 2015), 33 pages. https://doi.org/10.1145/2747874

[32] Yatao Yang, Biyu Ma, Jun Tan, Hongbo Deng, Haikuan Huang, and Zibin Zheng. 2021. FINN: Feedback Interactive Neural Network for Intent Recommendation. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1949–1958. https://doi.org/10.1145/3442381.3450105

[33] Lili Yu, Howard Chen, Sida Wang, Yoav Artzi, and Tao Lei. 2019. Interactive Classification by Asking Informative Questions. *arXiv preprint arXiv:1911.03598* (2019).