Towards an Optimal Dialog Strategy for Information Retrieval Using Both Open- and Close-Ended Questions

Yunfeng Zhang, Q. Vera Liao, and Biplav Srivastava
IBM Research
Yorktown Heights, New York, USA
zhangyun@us.ibm.com, vera.liao@ibm.com, biplavs@us.ibm.com

ABSTRACT

The emerging paradigm of dialogue interfaces for information retrieval systems opens new opportunities for interactively narrowing down users' information query and improving search results. Prior research has largely focused on methods that use a set of close-ended questions, such as decision tree, to learn about the user's search target. However, when there is a myriad of documents or items to search, solely relying on close-ended questions can lead to long and undesirable dialogues. We propose an adaptive dialogue strategy framework that incorporates open-ended questions at the optimal timing to reduce the length of the dialogue. We propose a method to estimate the information gain of open-ended questions, and in each dialog turn, we compare it with that of close-ended questions to decide which question to ask. We present experiments using several synthetic datasets designed to explore the behavior of such an adaptive dialogue strategy under different environments, and compare the system's performance with that of a close-ended-questions-only strategy.

ACM Classification Keywords

I.2.1 Artificial Intelligence: Applications and Expert Systems —*Natural language interfaces*; I.2.6 Artificial Intelligence: Learning—*Knowledge acquisition*

INTRODUCTION

We frequently engage in conversations to collaboratively narrow down information queries, for example, when a career consultant works with a client to identify the ideal job, when a salesperson helps a customer find a product, or when we discuss what kind of movie to watch together. We use diverse questioning strategies to make the conversation efficient and engaging. For example, the consultant may ask specifically "*what industry do you want to work in?*" because that would efficiently narrow down the choices. He or she may also ask open-ended questions like "*tell me what you care about work?*", in the hope of starting a reflective conversation where the client would reveal many preferences.

IUI'18, March 7–11, 2018, Tokyo, Japan

ACM ISBN 978-1-4503-4945-1/18/03…$15.00
https://doi.org/10.1145/3172944.3172998

There is a growing interest in adopting dialogue interfaces for information retrieval (IR) systems to take advantage of the interactivity that can improve the precision of information querying processes. In the simplest form, systems like Google Allo suggest query expansions for the next turn. Recent research (e.g., [7, 5, 9]) explored reasoning algorithms to ask follow-up questions, often by confirming additional features to use to narrow down the information query. While such a method is effective in improving IR results, it is inadequate for systems that aim to exhibit realistically human-like conversational behaviors. For example, to build a conversational agent taking the role of career consultant, the system should aim to identify one or a small number of candidate jobs for the user, instead of suggesting a long ranked list like conventional IR systems do. Relying solely on asking close-ended questions to confirm one feature at a time may also lead to long and boring dialogues from which the user is likely to drop out.

Towards building efficient and engaging conversational IR systems that adopt diverse questioning strategies like humans do, we propose an adaptive IR questioning strategy framework that alternates between asking close-ended and open-ended questions for feature elicitation, with the goal of minimizing dialogue length for identifying the user's target document or item. We present simulation experiments to compare the performance of the adaptive strategy to that of a baseline strategy that asks only close-ended questions. The simulations are set up to explore the performance of the adaptive strategy in several kinds of conditions characterized by (a) different user behaviors in answering open-ended questions, (b) different accuracies of the natural language processing system, and (c) different traits of the item-feature dataset (such as different mean values of features and correlations between features).

BACKGROUND AND RELATED WORK

With the recent development in NLP and conversational agents, conversational IR systems (e.g., search, recommender) are an emerging area of research drawing attention from both academia and industry [11]. Research has generally focused on two aspects. One is to develop methods to extract information from natural dialogue scripts to update queries or user models [2]. The other is to generate and manage the dialogues for information querying, e.g. [7]. A conversational interface is considered especially suitable for feature or preference elicitation for at least two reasons [1]. First, it is naturally interactive and thus users are likely more willing to provide additional information. Moreover, it can pro-actively and dy-

namically channel the information querying process in desired directions. For example, to narrow down users' queries, previous work explored asking confirmatory (Yes/No) clarification questions such as "*are you asking about topic X?"* [5], and eliciting values for a feature by direct inquiry (e.g., "*What VALUE would you like for FEATURE X?"*) or multi-choice questions (e.g."*choices for FEATURE X are VALUE 1... VALUE N"*) [3]. However, most existing work focused on system-initiative dialogues by asking close-ended questions. While it is intuitively efficient to obtain targeted information, it fails to consider the benefit of open-ended question such as "*what do you care about work?*" in eliciting a larger quantity of information, sometimes key preferences that the speaker is eager to provide.

IR QUESTIONING DIALOGUE STRATEGY FRAMEWORK

Our framework is based on a few assumptions about the IR dialog system. Firstly, we assume that there are at least hundreds of items to be searched with a large number of features. Smaller search spaces may not require a dynamic or lengthy dialog to identify the queried item. Secondly, we assume that when answering open-ended questions, users describe item features in natural language, and those features are extracted by a natural language processing (NLP) system. This means that the features are binary—they are either identified or not, but systems can accommodate features that have multiple values by converting them into multiple binary features. For example, the feature "color" can be converted into binary features that describe individual colors such as red, blue, etc. This mechanism is suitable for today's dialog systems that rely heavily on NLP techniques to extract relevant information.

The core idea of our framework is a question selector that optimally chooses between asking a close-ended question to inquire about a single feature and asking an open-ended question for which the user has a chance to report multiple features. The selector decides by maximizing the expected information gain (IG). While IG based approach has been explored in previous work [10] to decide what features to inquire about with closed ended-questions, we extend the consideration to open-ended questions. That is, at each dialog turn, the selector asks an open-ended question if the estimated IG_{open} of user answering an open-ended question is larger than the IGs of any close-ended question confirming a single feature.

The equation for calculating IG of a close-ended question for a given feature F is:

$$IG(F) = -\sum_{i=1}^{n} P(I_i) log P(I_i) + \sum_{i=1}^{n} \sum_{v=0,1} P(I_i) log P(I_i|F=v) \quad (1)$$

where n is the number of candidate items in the current state, $P(I_i)$ is the *a priori* probability that the item being queried is the ith item, and $P(I_i|F=v)$ is the conditional probability of I_i after knowing F's value. The IGs need to be estimated after every dialog turn for each unknown feature, because after filtering the candidate items by the feature(s) known from the last turn, the quantities in the above equations would change.

The above calculation of IG is similar to the one used by the ID3 decision tree learning algorithm in [10], which is only applicable to close-ended questions. To estimate the IG of open-ended questions, there are many factors to consider: the expected number of features that the user would report in answering the question λ_{NF} (a higher number represents a more cooperative user), the recall rate of the NLP system R_{nlp}, each unknown feature's information gain IG and likelihood to be reported L, and the correlations between features ρ. In situations where features have high degrees of correlation, ρ needs to be considered to avoid over-estimating the advantage of open-ended questions as the combined IG of multiple reported features would decrease. For simplicity, in this paper, we omit the consideration of ρ and leave it for future work. In the experiment, we will show that this omission does not have a huge impact if only a few features are correlated.

Given the above considerations, we propose that:

$$IG_{open} = R_{nlp}\, \lambda_{NF} \sum_{j=1}^{m} L_j IG(F_j) \quad (2)$$

Essentially, this equation states that the information gain of open-ended questions can be estimated as the weighted sum of the IG of all the m unknown features, multiplied by the expected number of features that would be reported and the NLP system's extraction rate.

Besides IG, all other parameters of Equation 2 can be either individually estimated from the data or jointly learned. For example, R_{nlp} can be measured by submitting testing data to the NLP system; λ_{NF} can be estimated by having human experts to count from historical data of user answers; and L can be learned from historical data by counting the report frequency of each feature. Reinforcement learning algorithms such as the one described by [8] can also be used to optimally allocate a number of open-ended questions to estimate the joint quantity of $R_{nlp} * \lambda_{NF}$. In this paper, however, we will not focus on the methods for estimating these parameters. Rather, we will assume that these estimations already have a reasonable degree of accuracy, and our primary goal is to explore the behavior of the adaptive strategy and compare its performance to a close-ended-questions-only dialog strategy.

SIMULATION EXPERIMENTS

We use a simulation approach to compare the performance of our adaptive IR questioning strategy with that of a baseline strategy asking only close-ended questions. Compared to running user studies, this approach allows us to quickly explore the behavior of the system in a wide range of conditions.

Setting up the Query Items Dataset

For the experiments, we simulate a user querying about one item out of n items with m features, where the goal of the dialog system is to elicit from the user the values of these features for the targeted item. We fix n to 1000 and m to 200 as they have little impact on the dialog strategy selection as accounted for by Equations 1 and 2.

The characteristics of the dataset that would most likely cause varying system behaviors are (a) the distribution of IG across features, and (b) the correlations between features. The distribution of IG would change how close the weighted average IG component in Equation 2 is to the highest IG of individual features, and hence alter how often the adaptive strategy

chooses open-ended questions. For correlation, high correlation between two features would reduce the combined IG of two features, which causes Equation 2 to overestimate IG_{open} since it assumes features are independent.

To vary the distribution of IG and the correlations across features, we went through three steps to generate the feature values of an item. Firstly, to determine Feature j's probability of being associated with an item, $P(F_j = 1)$ or $P(F_j)$ for short, we sampled a normal distribution $\mathcal{N}$ truncated to (0, 1) with mean μ and standard deviation σ. μ is fixed to a small value 0.05 to reflect the fact that each NLP-extracted concept or keyword tend to be associated with only a small portion of the items. σ changes the spread of $P(F_j)$ across features and hence the spread of IG (the closer $P(F_j)$ is to 0.5, the higher the IG), and we varied it between two values 0.3 and 0.9.

The second step of generating the feature values was to sample an m-dimensional multivariate-normal distribution $\mathcal{N}_m$ with feature dimension j's mean set to $P(F_j)$ and its standard deviation set to $\sqrt{P(F_j)(1-P(F_j))}$ (to mimic the binary Bernoulli distribution). By changing the covariance matrix of $\mathcal{N}_m$, we could vary the correlations between features. We used the hub-Toeplitz procedure (cf. [4]) to generate the correlation matrix. This procedure generates correlations that decrease from the maximum correlation ρ_{max} to the minimum correlation ρ_{min} at a specified rate γ. We set ρ_{min} to 0 and γ to 0.03 for all simulations such that only a few correlations are high and others are close to 0. We then varied the ρ_{max} between 0 (fully independent), 0.5, and 0.9 to explore the impact of feature correlations. After sampling $\mathcal{N}_m$, the last step was to convert the real-number feature values to binary by setting those below 0.5 to 0 (not associated with an item) and others to 1 (associated with an item). The above three-step procedure was then repeated n times to generate n items.

The Dialog Simulation Procedure

Besides the parameters for generating the query items, four other parameters impact the simulation. Three of them were about users' behavioral patterns: how likely they query each item $P(I_i)$, how likely they report each feature $L(F_j)$, and on average how many features they tend to report in answering open-ended questions λ_{NF}. Given a lack of user data to learn these reporting patterns, for $P(I_i)$ and $L(F_j)$, we explored a simple assumption where all features and items were equally likely to be reported and queried. For λ_{NF}, we explored a range of values between 1 and 3. Note that λ_{NF} used in Equation 2 is the expected mean value. When simulating the user, variations happen in each turn and occasionally the user can fail to report any feature existing in the system feature space. We simulated it through sampling the number of feature reported in a specific turn from a Poisson distribution with mean value of λ_{NF}. The one remaining parameter was the recall rate of the NLP system R_{nlp}, which was varied from low (0.2) to high (0.7). All parameters are summarized in Table 1.

As seen in Table 1, four parameters were varied: σ, ρ_{max}, R_{nlp}, and λ_{NF}. We ran one simulation experiment for each combination of their settings for a total of $2 \times 3 \times 3 \times 3 = 54$ experiments. Within each experiment, we ran 1000 trials, each querying about an item randomly drawn from the item dataset.

Name	*Description*	*Values*
n	Number of items	1000
m	Number of features	200
μ	Mean of $\mathcal{N}$ used to sample a feature's probability of being associated with an item	.05
σ	Standard deviation of $\mathcal{N}$	.3, .9
ρ_{max}	The maximum correlation used in the hub-Topelitz procedure	0, .5, .9
$P(I_i)$	Prob. of Item i being queried	uniform
$L(F_j)$	Prob. of Feature j being reported	uniform
R_{nlp}	NLP's feature extraction rate	.2, .5, .8
λ_{NF}	Mean of Poisson distrib. for generating the number of features reported in an answer to an open-ended question	1, 2, 3

Table 1. Summary of the simulation parameters.

Each trial went through many dialog turns until the target item was found. At each turn, the adaptive strategy picked the question with the maximum IG to ask, which could be an open-ended question if IG_{open} is larger than IG of any close-ended question. Next, we generated the simulated user answer. If the question was close-ended about Feature j, the feature value F_{ij} was taken as the answer, where i was the index of the target query item. If the question was open-ended, the simulated user first determined how many features it would report, N_r, by sampling a Poisson distribution with a mean of λ_{NF}. Then it randomly chose N_r features associated with Item i (feature value = 1) to report (to reflect the tendency that people report an item has a feature rather than lacks a feature). For open-ended questions, the answers were further passed through the simulated NLP system, where it sampled a Bernoulli distribution with a success rate of R_{nlp} to decide whether a feature would be extracted. The extracted features were then added to the known feature set to filter the candidate items for the next dialogue turn.

The Baseline Strategy and Performance Measures

We compared the adaptive strategy with a close-ended-questions-only baseline strategy. The questions of the baseline strategy were decided by SciPy's entropy-based decision tree classifier [6], which has the same performance as our proposed strategy when it only asks close-ended questions because both used the maximum information gain decision criterion. We considered two outcome measures from the simulation: the number of open- and close-ended questions asked in each trial. Next section compares the results of the two different strategies under the 54 different experimental settings.

Experimental Results

Figure 1 shows the results of the simulation. The bar length represents the mean of the total number of questions asked by the adaptive strategy in each trial. The green portion of the bar represents the number of open-ended questions asked, while the red portion close-ended. The horizontal line represents the number of questions asked by the baseline strategy, which remained at 10 across all conditions.

The results show that as λ_{NF} increased, i.e., users reporting higher number of features in answering open-ended question, the adaptive strategy asked more open-ended questions but

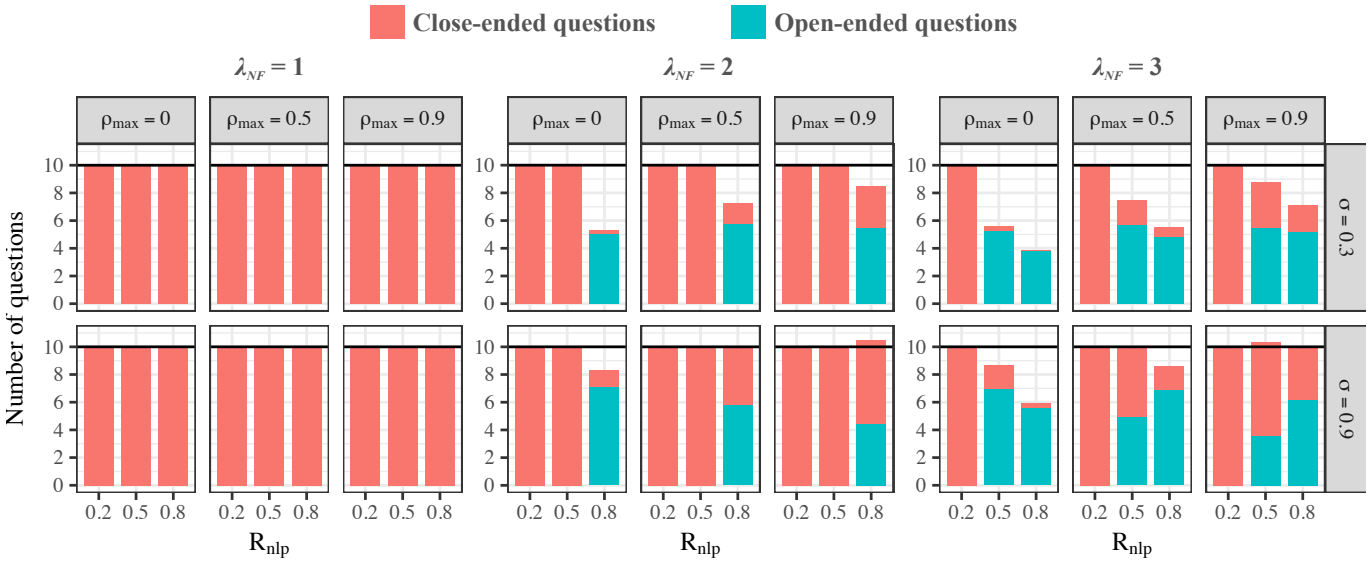

Figure 1. The average number of questions asked by the adaptive questioning strategy across the 54 experiments. The numbers are divided into close-ended (red) and open-ended (green) questions. The horizontal line at 11 of each panel represents the performance of the baseline strategy. Fewer questions, i.e. shorter bars, are better as they indicate fewer conversational turns are needed to determine the user's information need.

fewer questions in total. At $\lambda_{NF} = 1$, the adaptive strategy exclusively used close-ended questions, which resulted in similar performance to that of the baseline strategy. The adaptive strategy started to ask open-ended questions at $\lambda_{NF} = 2$ when the R_{nlp} is high, and at $\lambda_{NF} = 3$ when the R_{nlp} is medium or high. When open-ended questions were used, the total number of questions was reduced compared to that of the baseline strategy, demonstrating the effectiveness of our adaptive question selector. There are a couple of exceptions when $\sigma = 0.9$ and $\rho_{max} = 0.9$, and we will explore the reasons below.

The effects of σ and ρ_{max}, two parameters that affect the interrelations between features, are also clear from Figure 1. Comparing the first row and second row, we can see that as σ increased, the ratio of close-ended questions (red portion) increased. This is because high σ means more dispersed IGs across individual features, causing the maximum IG for close-ended questions more likely to be higher than the average IG used to derive IG_{open}. Therefore the adaptive strategy would shift to asking close-ended questions more often. Comparing the columns, we can see that as ρ_{max} increased, the total number of questions asked increased. This is because high ρ_{max} means high correlation between features, which would cause Equation 2 to overestimate the combined IG of multiple features, and hence overestimate IG_{open}. Due to this overestimation, the adaptive strategy would sometimes make suboptimal decisions and ask more questions in total. At the extreme conditions when ρ_{max} and σ were both high, the adaptive strategy even asked slightly more questions than the baseline strategy, as can be seen in Figure 1 where two bars exceeded the horizontal line.

DISCUSSION AND CONCLUSION

The simulation results suggest that under most conditions, because of the ability to exploit open-ended questions, the adaptive strategy would need to ask fewer questions than the decision-tree-based, close-ended-questions-only strategy. The results suggest that in ideal conditions such as when the users are cooperative in describing multiple features in answering open-ended questions, and when the accuracy of the NLP system is high, the adaptive strategy could slash the number of questions by half or even two-thirds. Since the number of dialog turns highly affects user satisfaction [12], our proposal could substantially increase the usability of a dialog system.

Several limitations of this study need to be addressed in future research. Firstly, to apply the adaptive strategy in conditions where the correlations between features are high, we need to improve Equation 2. Potential methods include calculating and averaging the IGs of all possible λ_{NF}-feature combinations. This however may be computationally intractable when there is a large number of features or when λ_{NF} is high. Secondly, as suggested previously, integrating a reinforcement learning algorithm in our adaptive strategy could help learn the expected number of features extracted by the NLP system from the historical data. Lastly, the same or similar learning mechanisms would also be useful for learning the probability that each item is to be queried and that each feature is to be reported. Knowing these parameters would further optimize the performance of our proposed system.

In conclusion, we showed that our method of including open-ended questions in an IR dialog system can substantially increase the efficiency of a dialog and our proposed method for estimating the information gain of open-ended questions can account for many critical factors, which helps our adaptive strategy make near optimal decisions about when to ask open-ended questions and when close-ended.

ACKNOWLEDGMENTS

We are indebted to Stefan Liesche for his enthusiastic support of this research, to Matthew Davis for his leadership of the overall project, and to Rachel Bellamy for her comments and feedback on the manuscript.

REFERENCES

1. Susan E Brennan. 1990. Conversation as direct manipulation: An iconoclastic view. (1990).
2. Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems.. In *KDD*. 815–824.
3. Jennifer Chu-Carroll. 2000. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. In *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics, 97–104.
4. Johanna Hardin, Stephan Ramon Garcia, and David Golan. 2013. A method for generating realistic correlation matrices. *The Annals of Applied Statistics* 7, 3 (sep 2013), 1733–1762. DOI: http://dx.doi.org/10.1214/13-AOAS638
5. Eric Horvitz and Tim Paek. 1999. A computational architecture for conversation. In *UM99 User Modeling*. Springer, 201–210.
6. Eric Jones, Travis Oliphant, Pearu Peterson, and others. 2001. SciPy: Open source scientific tools for Python. (2001). http://www.scipy.org/
7. Kazunori Komatani, Tatsuya Kawahara, Ryosuke Ito, and Hiroshi G. Okuno. 2002. Efficient dialogue strategy to find users' intended items from information query results. In *Proceedings of the 19th international conference on Computational linguistics*, Vol. 1. Association for Computational Linguistics, Morristown, NJ, USA, 1–7. DOI:http://dx.doi.org/10.3115/1072228.1072380
8. T.L Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* 6, 1 (mar 1985), 4–22. DOI: http://dx.doi.org/10.1016/0196-8858(85)90002-8
9. Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126* (2016).
10. J.R. Quinlan and J. R. 1986. Induction of Decision Trees. *Machine Learning* 1, 1 (1986), 81–106. DOI: http://dx.doi.org/10.1023/A:1022643204877
11. Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*. ACM, 117–126.
12. M A Walker, D Litman, C A Kamm, and A Abella. 1997. PARADISE: A general framework for evaluating spoken dialogue agents. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, ACL/EACL 97* (1997), 271–280. DOI: http://dx.doi.org/10.3115/979617.979652