

SMARTREC - A SMART CONVERSATIONAL RECOMMENDATION SYSTEM
USING SEMANTIC KNOWLEDGE GRAPHS

A Thesis

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Sudha Vijayakumar

May 2022

© 2022

Sudha Vijayakumar

ALL RIGHTS RESERVED

The Designated Thesis Committee Approves the Thesis Titled

SMARTREC - A SMART CONVERSATIONAL RECOMMENDATION SYSTEM
USING SEMANTIC KNOWLEDGE GRAPHS

by
Sudha Vijayakumar

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2022

Dr. Jorjeta Jetcheva Department of Computer Science

TBD Department of Computer Science

TBD Department of Computer Science

ABSTRACT

SmartRec - A Smart Conversational Recommendation System using Semantic Knowledge Graphs

by Sudha Vijayakumar

Conversational recommendation systems (CRS) intend to return the best recommendations to users through a multi-turn interactive conversation. CRS allows users to provide their feedback during the conversation, unlike the traditional recommendation systems. CRS can combine the knowledge of the predefined user profile with the current user requirements to output custom yet most relevant recommendations or suggestions [1]. The Covid-19 pandemic has undoubtedly accelerated the pace of the adoption of conversational AI systems by many e-commerce platforms to provide highly available customer support. This ongoing demand calls for the need to implement more generic and efficient conversational agents and recommendation engines that can provide customers with the required information at every stage of the conversation during the purchasing and issue-resolving cycle. This study will implement a Smart Conversational AI-based recommendation system - SmartRec. SmartRec can have a multi-turn conversation with the user to understand the context and semantics behind their product requirement or issue reporting. SmartRec can generate appropriate recommendations or natural language response text suggestions based on the user queries. Although the diverse study is in progress for improving CRS [3], there are multi-faceted gaps that remain open for research. State-of-the-art CRS suffers from three main issues. first, lack of proper contextual understanding of the user inclinations. (e.g., what is the current user goal when booking a vacation); second, inaccurate semantic mapping of user preferences in natural language to the interested item attributes; third, rely only on current conversation and suffer from data sparsity [4] by not incorporating

historical user interactions for understanding the user profile. To overcome these issues, SmartRec will encompass a context-aware semantic knowledge graph that captures the current conversation, past conversations, and product attributes to produce relevant and semantically meaningful responses to user queries. On top of this intelligent knowledge graph, this study will integrate a collaborative recommendation framework [5] and a dialog component [6]. This study will conduct experiments to evaluate machine learning, system performance, and user experience and compare them against best-in-class CRS [7]. State-of-the-art knowledge graph-based CRS's are trained on large-scale multi-domain data but do not generalize well for a specific domain. In addition, they lack features that can assist interactive conversation and do not include historical user-item interactions. This work envisions building a novel CRS, utilizing semantic knowledge graph and machine learning components by incorporating a hybrid response model, which combines the ability to list product recommendations with suggestions for the reported customer issue. This work will curate a novel large-scale, domain-relevant, integrated dataset from different sources such as Airbnb tagged data from Twitter and Quora, public data released by Airbnb, including listings, user reviews, and FAQs for experiments. Our CRS will incorporate preference understanding by current context, historical user interactions, and user engagement. The intent of this work is to propose a system architecture that is generic and thus easy to train with data for different domains and adapted to a broad range of AI-enhanced customer service applications. The current study will be limited to the textual conversation in English. Future extensions of this solution can incorporate image and video interactions, voice-based, and multilingual CRS, expanding the potential footprint of the solution to an even a broad range of applications and users.

ACKNOWLEDGMENTS

I want to thank my Professor. Marie Highby for guiding me throughout this coursework.

TABLE OF CONTENTS

CHAPTER

1	INTRODUCTION	1
2	RELATED WORKS	5
3	PROJECT JUSTIFICATION	8
4	TECHNICAL ASPECTS	9
5	PROPOSED APPROACH	10
5.1	Modeling NLU Data	10
5.2	Modeling NLG Utterances	11
5.3	Modeling the Dialog Manager	12
5.4	Train the Model	12
5.5	Recommendation engines	13
5.5.1	Offline vectors and models	13
5.5.2	Real-time graphs and models	14
6	EXPECTED RESULTS	15
7	EXPERIMENTS	16
8	CONCLUSIONS AND FUTURE WORK	17

CHAPTER 1

INTRODUCTION

A Forbes article states that 72 percent of e-commerce users prefer to seek answers online. By 2022, according to Gartner reports, 70 percent of middle-class workers will utilize conversational stages each day [1]. 58 percent of consumers say that they care even more than before about the customer experience in the post COVID world [2]. The Covid-19 pandemic has undoubtedly accelerated the pace at which e-commerce platforms will adapt conversational agents for highly available customer support and, Gartner forecasts second that. This projected demand calls for the need to implement more generic and efficient conversational agents and recommendation engines that can provide customers with the required information at every stage of the conversation during the purchasing and issue-resolving cycle. The need for a conversational recommender framework (CRS) has turned into an arising research point in trying to furnish quality recommendations through interactive discussions with users. The ability of the conversational recommendation system to allow users to express interests in natural language dialects exceptionally affects e-commerce business.

As far as technique, CRS requires consistent incorporation between a recommender part and a dialog part. The dialog part explains user intentions and answers to the past expression with reasonable responses. Then again, the recommender part learns user inclination and suggests closely matched things dependent on context-oriented natural language expressions. Traditional conversational recommendation systems incorporate two core components: 1) a conviction policy agent to validate partially formatted user requirements and 2) a switching controller to choose between the dialog component and the recommender component.

Albeit these examinations have worked on the exhibition of CRS somewhat, two significant issues still need to be addressed. Initially, a discussion comprises a couple of sentences, absence of adequate relevant data for precisely understanding user inclination. As displayed in Table 1, a client is searching for peaceful and quiet Airbnb places like "Serene ocean side house", where the user's inclination is essentially portrayed by fewer sentences. The context of the conversation has to be understood to capture the user's interests accurately. In the stated example, it is imperative to capture the hidden semantics of "safe" and the Airbnb posting "Serene ocean side house". first, Capturing reality information solely subject to articulation utterances is a cumbersome task; second, authentic user tendency is mostly reflected over the objects or emotions (e.g., locality, experience) whereas articulations are tended to in ordinary lingos. So, it is natural that semantic voidness exists between the information signals required to provide recommendations such as the way the user expresses the requirements and the way items are defined. Therefore, a powerful semantic combination method for comprehension or creating combined expressions is required. As displayed in Table 1, the framework has introduced both the suggested listing and the justification behind the proposal. Without connecting the semantic hole, it is infeasible to determine the user intent for coming up with relevant suggestions, e.g., the user intent in natural language expression such as "a safe beach house" may be mapped to a listing with the description "serene ocean-side house." They mostly center around joining item information, while the word-level advancement (e.g., semantic connection between the texts, safe and serene) has been some way or another disregarded. Besides, they have not thought about the semantic hole between user language and outside information. Accordingly, the usage of Knowledge Graph information is probably going to be restricted. The issue starts right from the way the dialog part and the recommender part are aligned with

entirely different semantic grounds based on user utterances and descriptions of the item attributes. Our thought is to fuse the item-level and the word-level attributes using a semantic item-based Knowledge Graph and a semantically trained NLU, NLG machine learning models for mapping user requirements to the available items and coming up with semantically meaningful item recommendations.

This proposal presents SmartRec, a Smart Conversational AI-based recommendation system. SmartRec can have a multi-turn conversation with the user to understand the context and semantics behind their product requirements. SmartRec can generate appropriate recommendations/ natural language response text suggestions based on the user queries. Most of the state-of-the-art CRS are domain-specific, lack meaningful multi-turn conversation, lack contextual understanding of the user preferences, and possess a semantic gap between user intent in natural language and product attributes. This work proposes a solution that offers (1) an auto-suggest type-ahead mechanism [5] to help users convey their intent driving for more meaningful conversations (2) a semantically trained machine learning model that captures the context of the user intent in natural language and queries the semantic item knowledge graphs to the return corresponding products and (3) an integrated recommendation and dialog engine to generate product and solution recommendations using the current user interactions. The goal is to fuse current data from the user conversation, historical user-item interactions, historical issue handling data, and social-conversational data of a particular brand to better understand the user, match and, return the products/ solutions that interest them. In light of the adjusted semantic portrayals, this work further fosters a real-time Knowledge Graph-based recommender module for making exact suggestions and a semantically trained language model that can understand the semantics behind the interactive conversation with the user and come up with semantically meaningful responses. As

far as anyone is concerned, it is the initial occasion when the coordination of discourse and recommender frameworks has been tended to by utilizing Knowledge graphs and semantically trained language models. Broad examinations on an open-source Airbnb dataset have shown the viability of the proposed approach in both recommendation and conversational tasks in the e-commerce domain.

CHAPTER 2

RELATED WORKS

A typical architecture of a conversational AI-based recommendation system (CRS) shown in Figure 1 consists of two core components, namely the recommendation engine for generating appropriate recommendations and a language model for understanding user queries and generating responses. The below section will discuss some interesting related works in both aspects.

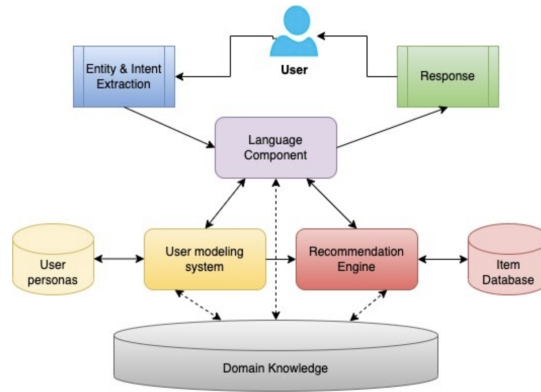


Figure 1: Typical architecture of a conversational recommendation system.

Recommender frameworks expect to distinguish a part of the itemset that satisfies the user requirements from a dense item space. Customary techniques are exceptionally founded on the recorded user ratings for the itemsets (e.g., rate, click, number of views, buy). Be that as it may, user-item collaboration information is normally meager. To handle the information sparsity issue, numerous methods have been created by using the side data of things, for example, survey and scientific categorization information. As a comparable approach, CRS predominantly centers around the recommendation setting through current discussions rather than authentic collaboration information. Particularly, many novel recommendation solutions are beginning to embrace knowledge graphs to enhance the performance of the recommendation task and semantic logic of the explainability.

Conversational AI frameworks aim to produce appropriate reactions given multi-turn context-oriented expressions. Existing works can be classified either into generative or retrieval-based methods. Generative-based approaches use learnable models to create the response text whereas the retrieval-based methods attempt to come up with the semantically meaningful response from a huge store of historical user discussions. With the improvement of sequence-sequence language processing transformer models, various architectural extensions have been implemented to understand language utterances and generate useful responses back.

Traditional conversational recommendation engines essentially used predefined functions to communicate with users. As of late, a few investigations began to coordinate the two parts for decoding user intentions and suggest the appropriate items through regular natural language articulation. Generally speaking, these techniques underscore the exact proposal, while the discussion part is carried out by straightforward or heuristic arrangements. In most cases, a crowd sourced conversational dataset has been used, and a progressive neural network model was trained to generate responses. Moreover, further examinations joined diverse Knowledge Graphs to work on the CRS, where the main focus was to principally improve the identification of items.

The baseline paper for this work, KGSF proposes a unique conversational recommendation system by incorporating and combining crowd sourced conversational data word-level and open domain entity-level knowledge graphs. With the fusioning of knowledge graphs, the KGSF models can learn data representations more cohesively for a better recommendation and response generation. However, combining semantic embeddings of knowledge graphs is an expensive task which requires hours of data preprocessing and training the model. Also, using open domain data can add too much variance to the resulting model and crowd sourced

conversational data could add bias to the overall result.

Based on these past research works, we propose to design and implement a novel conversational recommendation system using item-based semantic knowledge graphs and word-level Wikipedia semantic map embeddings. The proposed work will also generate the conversational data automatically using the available item entities for three different pre-defined user personas. An NLU and an NLG pipeline shall be incorporated to extract entities and identify user intents from the user interaction and come up with appropriate responses for user queries. The recommendation part of the proposed solution will be based upon a hybrid pre-trained and real-time collaborative and content-based filtering approaches.

CHAPTER 3

PROJECT JUSTIFICATION

Through our work, we propose to implement an expert CRS using publicly available large-scale Airbnb data. The large-scale Airbnb data presents a substantial modeling challenge for our proposed study and stands as a representative of a broad range of use cases involving consumer choices and decision-making. In turn, this shall reflect the notion that the models resulting from our work will have applicability for customer support in the context of similar e-commerce or vacation planning applications. We plan to leverage Random Access Navigation(RAN) [8] type-ahead approach-based UI to drive a freestyle yet guided conversation for the user. The type-ahead approach allows users to converse with guidance and make corrections anytime as humans act in real life. Probabilistically, RAN only looks for missing parameters and then asks only relevant questions eliminating the need to hard-code all the possible $n*(n-1)$ cases for the given n parameters. A good product recommendation or customer support system relies upon an accurate interpretation of the user preferences. Therefore, we propose to leverage the semantic fusion approach to contextually map the user preferences to the available products or suggestions while remembering the collective requirement from the chat history. A word-oriented semantic embedding like [13] (i.e., ConceptNet3, Wikipedia) using Airbnb’s custom product-attribute knowledge graph using Airbnb’s item listings, historical user interactions, and user reviews will add data intelligence to the system. We plan to include a real-time and a pre-trained neural collaborative filtering framework [6] to implement an integrated recommendation cum dialog component [7] that will act as retrieval and generative response model to list the relevant recommendations and response text. We intend to implement a generic solution architecture for an AI-enhanced product recommendation and customer support systems in e-commerce.

CHAPTER 4

TECHNICAL ASPECTS

State-of-the-art knowledge graph-based CRS's [4] trained on large-scale multi-domain data but do not generalize well for a specific domain [13], lack features that can assist interactive conversations, and do not include historical user-item interactions. We envision building a novel CRS utilizing semantic knowledge graph and machine learning components, incorporating a hybrid response model which combines the ability to list product recommendations with suggestions for the reported customer issue. As part of our research, we will curate a large-scale, domain-relevant, integrated dataset from different sources such as Airbnb tagged data from Twitter and Quora, public data released by Airbnb, including listings, user reviews, FAQs. Our CRS will incorporate preference understanding by current context, historical user interactions, and user engagement.

We intend our proposed system architecture to be generic and thus easy to train with data for different domains and adapted to a broad range of AI-enhanced customer service applications. Future extensions of this solution will incorporate image and video interactions that accompany the conversation, voice-based, and multi-lingual CRS [9], expanding the potential footprint of the solution to an even a broad range of applications and users.

CHAPTER 5

PROPOSED APPROACH

The proposed solution shown in Figure 2 will leverage the RASA conversational development platform with four core elements: NLU model, Dialog State Manager, NLG model, and Recommendation engine: 1) An NLU model to grasp the semantics behind the interactive conversation and derive the expectation of the user from natural language expressions, 2) A Dialog State Manager will keep track of the conversations and trigger corresponding action points for conducting the chain of actions through custom APIs, 3) An NLG component to generate appropriate responses to be conveyed to the user and 4) a recommendation engine to fetch top recommendations for the provided user query and communicate via the NLG component.

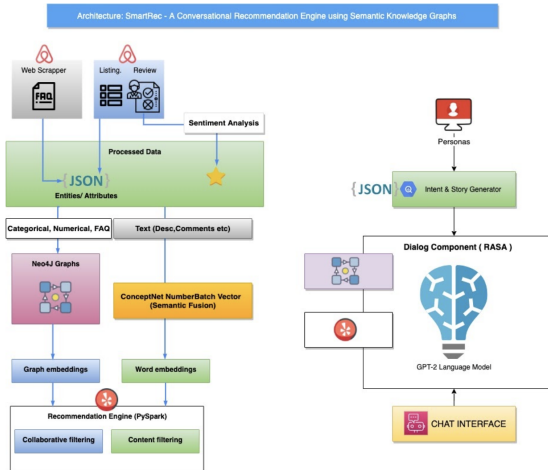


Figure 2: SmartRec Conversational Recommendation System using RASA and Neo4j.

5.1 Modeling NLU Data

This work will implement a closed-domain conversational AI system for the e-commerce platform with models trained using Airbnb open-source data. Pre-defined expected intents, sentence structures, and entity types will act as the training data

for the NLU model. Figure 3 is an example of a sample NLU model intent sentence structures without any entity types. Every entity type has a name and value.

Figure 4 is an example of a sample NLU model intents, sentence structures with entity types defined. This work aims at auto-generating the intent data, sentence structures, and entity types using the Airbnb domain data. This work will model around ten entity types from 10,000 data samples. The total number of intents will be approximately around 75 (25 intents per user persona).

```
- intent: greet
  examples: |
    - Hello there
    - Hello
    - Hi
    - Hey
    - Are you there?
```

Figure 3: Sample intent without entities.

```
- intent: query_knowledge_base
  examples: |
    - What nice [Listing](object_type) is there?
    - Is there any good [Listing](object_type)?
    - List me some [Listing](object_type)
    - List me some [State](object_type)
```

Figure 4: Sample intent with entities.

5.2 Modeling NLG Utterances

The NLG module will be trained using varied intents followed by different but synonymous conversational agent responses. That is to replace stereotypical responses in traditional recommendation engines with a more natural way of conversing. Fig 3. shows an example of different responses a conversational agent generates for a given intent.

5.3 Modeling the Dialog Manager

The dialog manager will be modeled using three distinct Airbnb user personas using RASA stories and domain configurations. RASA stories model the conversational flow for every user and enable multi-turn conversation. An Intent classification algorithm integrated with the dialog manager can identify intents from the current context of the conversation and trigger corresponding actions from the conversational flow. For every user intent, the conversational agent responds with appropriate Airbnb recommendations. A massive amount of model data assists the dialog manager with anticipating and performing activities all the more precisely. Fig 4 shows an illustration of a sample dialog. In Fig 4, the (explain the example).

5.4 Train the Model

This work will implement conversational AI using the RASA framework. The model pipeline to be implemented is shown in Fig 5. The input text and features are tokenized and vectorized using WhiteSpaceTokenizer and BERT model featurizer. NLU models trained to understand the semantics behind the interactive conversation using the pre-trained Wikipedia semantic mappings. NLG models trained using the DIETClassifier to classify the user intents appropriately to enable the dialog manager to trigger relevant action APIs. The EntitySynonymMapper is used to map the user conversation with synonymous words.

The NLU data intent class names and the sentence structures are converted to numerical, after which the sentence structures of the intents are tokenized by whitespaces and transformed to vectors using WhiteSpaceTokenizer. The vectorized data provided to the Dual Intent Entity Transformer (DIET) classifier for intent classification and entity extraction. The pre-defined intents, entities, slots, and stories are transformed into a binary vector to match the size of the sum of total

intents, entities, slots, and former actions. The expected action is vectorized using a one-hot encoder equal to the size of the sum of the actions. The vectorized data is fed to the BERT language model for further training.

The following policies are used to decide the action to be triggered depending upon the identified entities and user intent.

5.5 Recommendation engines

5.5.1 Offline vectors and models

A hybrid neural recommendation model shown in Figure 5 is trained using content-based and collaborative-based approaches on the Airbnb user-item rating dataset. This trained model can be used offline to generate recommendations for the specified user queries using the defined model policies shown in Figure 6. The approximate training time for 5.5k itemset and 255k user-item ratings is around 30 minutes. Feature vectors of the Airbnb listings data and user review data are created for offline querying for the available listings.

```
language: en
pipeline:
- name: WhitespaceTokenizer
- name: LanguageModelFeaturizer
  model_name: bert
  model_weights: "rasa/LaBSE"
- name: RegexFeaturizer
- name: rasa_nlu_examples.featurizers.sparse.SemanticMapFeaturizer
  pretrained_semantic_map: "data/rasa-sme-wikipedia-en-128x128-v20201120.json"
- name: DIETClassifier
  epochs: 1000
  learning_rate: 0.001
- name: EntitySynonymMapper
- name: FallbackClassifier
  threshold: 0.2
  ambiguity_threshold: 0.1
```

Figure 5: SmartRec model pipelines.

```
policies:
- name: MemoizationPolicy
- name: TEDPolicy
- name: RulePolicy
  core_fallback_threshold: 0.2
  core_fallback_action_name: action_default_fallback
```

Figure 6: SmartRec model policies.

5.5.2 Real-time graphs and models

Airbnb listings and user reviews are recorded into the Neo4j graph database using bulk admin import. Preprocessing took approximately 6 minutes and Neo4j graph data upload took approximately 8ms for 5.5k itemset and 255k user-item ratings. This work will implement a real-time querying interface of the Airbnb listings and a hybrid graph-based real-time recommendation using collaborative cum content-based filtering approaches. The vector-based recommendation approach requires a massive quantity of data to overcome the cold-start issues like few users and few itemsets, which require intensive training. Real-time recommendation systems using knowledge graphs are fast, up-to-date, eliminate the need for pre-trained embeddings, support semantic relationships, and can effectively handle cold-start issues.

CHAPTER 6

EXPECTED RESULTS

Big e-commerce players like Airbnb, Amazon are researching efficient ways to enhance customer experience and provide a highly available AI-enabled customer support system. This study intends the proposed real-time conversational recommendation system to be a 24x7 customer support system for e-commerce platforms. This study will focus on an Airbnb-inspired use case for which a large amount of data has been made available to the public by the company. Airbnb has implemented a task-oriented customer support bot [12] to provide customer support during the COVID pandemic and has added flexible filters to tackle the rigidity of the current product search system. SMARTRec has the potential to significantly expand these existing capabilities with a much more sophisticated AI-based approach.

CHAPTER 7

EXPERIMENTS

This study will conduct extensive experiments to evaluate the conversation task and the recommendation task individually and after integration based on different evaluation criteria. Evaluation metrics include intent classification accuracy, number of successful multi-turn conversations, and qualitative analysis of the system performance compared to the best-in-class conversational recommendation systems. The model performances of the initial experiments conducted as tabulated in Table 1

Table 1: Model performance

Case	Name	Percentage
1	Naive Bayes Classifier	87%
2	DIET Classifier	92%

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

This paper proposes a unique real-time Knowledge Graph-based semantically intelligent interactive conversational recommendation engine to recommend top matched items to the user. The proposed system will leverage the RASA development platform and a graph-based recommendation engine using Neo4j. The knowledge graph enhanced recommendation and querying interface is expected to outperform the performance of the vector-based recommendation frameworks. Furthermore, the semantic understanding is included in the NLU model as a pre-trained common sense vector eliminating the need to train from scratch. This approach will save hours of training time required to combine the item data and the common-sense data into one unified semantic dataset. This study will incorporate only relevant historical user reviews on the itemset and the item entities to conduct the conversation with a pre-defined user personas with domain-relevant auto-generated mock conversational data instead of unspecific crowd-sourced conversational data.

The extensions of this work will be adaptable to many e-commerce platforms that include big players like Airbnb, Amazon. The system will help the e-commerce players improve customer experience and provide an AI-enabled customer recommendation and support system. This study will focus on an Airbnb-inspired use case for which a large amount of data has been made available to the public by the company. Airbnb has implemented a task-oriented customer support bot [12] to provide customer support during the COVID pandemic and has added flexible filters to tackle the rigidity of the current product search system. SMARTRec has the potential to significantly expand these existing capabilities with a much more sophisticated AI-based approach.