SSCR (Self-Supervised learning for Conversational Recommendation) designed to overcome two **key challenges**:1. the need for vast amounts of training data and 2.the failure to adequately utilize the diverse knowledge derived from dialogues.

**Solution**:The SSCR model employs self-supervised learning to identify connections between different types of knowledge. It uses **three self-supervision signals** and **two auxiliary self-supervised objectives** at token and sentence levels to unearth semantic knowledge. It then applies **external knowledge graphs** to extract structural knowledge from user-mentioned entities. It also uses **contrastive learning** to model interplay between semantic and structural knowledge. A **new similarity function based on negative log-likelihood loss** is proposed as existing ones were ineffective.

Introduction:

CRS methods face two significant issues: **data sparsity** due to the time-consuming nature of collecting quality recommendation samples, and **insufficient extracted knowledge** from dialogues due to limited user interaction turns.

SSCR uses self-supervised learning (SSL) to tackle the data sparsity and knowledge scarcity.

**three self-supervised tasks** designed to learn **semantic knowledge(**derived from user-mentioned words**)**, **structural knowledge(** derived from user-mentioned entities from user interactions**)**, and their **cross-knowledge**.

1. For the **semantic knowledge**, we propose a **token-level task** and a **sentence-level task**.

1.token-level task: we consider to model the intrinsic correlations of words in both the recommendation module and dialogue module. This task models the semantic features of words, and captures user preferences more accurately.

2.sentence-level: semantic relevance extraction task, it brings the contrastive learning (CL) framework to improve the fluency and coherence between sentences.

1. The **structural knowledge** is **modelled from the intrinsic correlations of user mentioned entities** via knowledge graphs (KGs). For example, ‘‘horror film’’ and ‘‘The Shining’’ are one-hop neighbours under the relation ‘‘movie type’’, and are prone to appear in the same dialogue. Thus the structural knowledge of entities contributes to capturing the user’s taste, and enhances the accuracy and rationality of recommendation.

(3) The **cross knowledg**e refers to the inter-information between semantic words and structural entities. For example, when people talk about the horror film ‘‘The Conjuring’’ (entity), they may also say ‘‘scary’’ and ‘‘creepy’’, rather than ‘‘funny’’ or ‘‘comedy’’. Thus, we view the words and entities as two views of user representations, and learns the inter-information with the advantage of contrastive learning framework. Concretely, it maximizes the agreement between the entities and words in the same dialogue session, compared to that in the different sessions. To better achieve this goal, we propose a novel similarity function based on the **negative log-likelihood loss**, rather than the inner product. Besides, we propose a **new generator** for better incorporating the learned user representations.

1. Preliminary and formulations

2.1. Notations and problem statement  
A modern CRS consists of two primary modules: a recommendation module and a dialogue generation module.

From the dialogue with the users, we collect the user mentioned structural entities e ∈ E and semantic words w ∈ W. Here E and W denote the total entity set and word set, respectively. **The recommender system** aims to capture the user intents from the words and entities, and retrieves the user’s favourite items from the candidate pool.

For **the dialogue generation module**, it aims to generate fluent and accurate response utterances given the T − 1 turns of dialogue history . When generation, it automatically decides to recommend an item or raise a question/chit-chat response. If the first case, it calls the **recommendation module** to provide proper items.

2.2. Base CRS model

2.2.1. **Recommendation module**

1.Assume that the user u provide m semantic words 𝑢 = {𝑤1 , …, 𝑤𝑚} and 𝑛 structural entities 𝑢 = {𝑒1 , … , 𝑒𝑛 }. We encode them as user representations.

2.To learn the semantic representations of user mentioned words, we adopt the ConceptNet as semantic knowledge and leverage (GCN) as the representation extractor.

1. With the message passing and aggregation, we derive the **semantic matrix S** ∈ R𝑛𝑠×𝑑 , which is the semantic representations of the whole 𝑛𝑠 words.
2. Then we lookup the embeddings S𝑢 = {𝑠𝑤1 , …, 𝑠𝑤𝑚 } of user mentioned semantic words W𝑢 for recommendation.
3. For user mentioned structural entities, we leverage DBpedia to learn structural representations. It stores structural facts < 𝑒1 , 𝑟𝑠𝑡, 𝑒2 >, where 𝑒1 , 𝑒2 ∈ E are item or non-item entities from the entity set E and the 𝑟𝑠𝑡 ∈ R is the relation between entities from the relation set.
4. As the entity relation is a useful signal, here we adopt R-GCN to learn the structural matrix:

一張含有 字型, 文字, 白色, 行 的圖片

自動產生的描述（1）

where is the 𝑙th layer’s representation of  are trainable parameters. With Eq. (1), the **user mentioned structural entities**  are embedded into **entity representations**.

As users may mention different numbers of entities, 𝑛 is not a fixed value. For convenience,

7.we incorporate the self-attention mechanism (𝑓𝑡 ) to average T𝑢 according to the importance of entities:

一張含有 字型, 文字, 印刷術, 書法 的圖片

自動產生的描述（2）

where  is a linear transformation,  are learnable parameters, and is the **entity representation.**

8. Similarly,  (semantic martics) is also averaged by the self-attention 𝑓𝑠 to derive the **word representations**

9.Next , we leverage the **gate mechanism** to fuse the two parts and derive the **user representation 𝒓𝑢** :

一張含有 字型, 書法, 筆跡, 印刷術 的圖片

自動產生的描述（3）

where 𝜎 is Sigmoid function.

10. Finally, the probability that user likes the item 𝑖 is calculated by:

（4）

where T denotes the structural matrix that includes **the representations of all entities**, according to Eq. (1), and [𝑖] means the 𝑖th element.

**2.2.2. Generation module**

We adopt the classical seq2seq framework for dialogue generation, which includes an encoder and a decoder. Concretely, we use **Transformer** as the base model for dialogue encoding. Generally, the standard Transformer architecture consists of several sub-layers that have a multi-head attention layer and a fully connected feed-forward layer. Then a **residual connection** and **normalization layer** around each two sub-layers

3. Method

**3.1. Overview**

SSCR uses self-supervised learning (SSL) to better model user preferences through the fusion of semantic and structural knowledge.

The proposed approach involves three main self-supervised tasks: **semantic knowledge extraction, structural knowledge extraction, and cross knowledge extraction**.

1. For **semantic knowledge extraction**, it designs a **token-level task** to reflect the correlations among **user-mentioned words** and a **sentence-level task** to enhance fluency and coherence between **context history** and generation utterances.

2. For **structural knowledge extraction**, it utilizes the inherent correlations among **user-mentioned entities** and leverages the **structural knowledge** from DBpedia.

3. Finally, **cross knowledge extraction** targets the interplay between semantic and structural knowledge, leveraging co-occurrence relations between semantic words and structural entities.

To effectively extract the **cross-knowledge**, SSCR employs a **contrastive learning framework** and introduces a novel **negative log-likelihood (NLL) loss-based similarity function**. Additionally, a **knowledge-aware generator** is proposed to incorporate the three types of knowledge in dialogue generation more effectively. The SSCR method is shown to provide a more complete and robust approach to understanding and utilizing user knowledge in CRS.一張含有 文字, 螢幕擷取畫面, 圖表 的圖片

自動產生的描述

3.2. Semantic knowledge extraction

3.2.1. Modelling semantic features of words

In this section, we introduce the **words’ semantic feature extraction task**. Words Semantic Task for Recommendation. In each dialogue session, the user mentioned words have fine-grained connections. For example, when users mention ‘‘scary’’ and ‘‘horror’’ in dialogue history, it can be inferred that they are talking about horror films, and the user is prone to say ‘‘thriller’’ in the following, rather than ‘‘romantic’’. Thus, the intrinsic correlations of words contribute to understanding the user preferences. So, the **first task** is to discriminate the target word given the surrounding context words. Concretely,

1.we **first randomly delete the 𝑘th word** and then leverage the rest words  to predict the masked item 𝑤𝑘 :

（5）predict mask掉的word来加强对语义的理解

where (通过GCN把W提炼后的representation) is the **word representation**s of  , **𝑓𝑠 is the self-attention** mechanism according to Eq. (2) and **S** ∈ R𝑑×|| denotes the **semantic representations of the whole word set**. Then we minimize the Semantic Features of Words (SFW) loss for recommendation by:

（6）

Similarly, for the **dialogue generation task**, we also encourage model to be aware of the semantic features and semantic connections among words. Specifically,

1.given the **user mentioned context utterance** 𝑋𝑢 = {𝑥1 , …, 𝑥𝑘 ,… , 𝑥𝑝 } with 𝑝 tokens, we randomly mask the 𝑘th token and derive masked surrounding context = {𝑥1 , …, [MASK],… , 𝑥𝑝 }. Then the masked token 𝑥𝑘 ∈  is predicted as:

一張含有 字型, 文字, 白色, 書法 的圖片

自動產生的描述（7）

where **Emb𝑐 refers to the embedding layer** of the Transformer Encoder, Pooling is the average pooling of tokens’ embedding, 𝑑𝑡 is the dimension of word embeddings and 𝑊𝑐 ∈ R𝑑𝑡×|V| is a trainable linear projections that transforms 𝒄 into |V|-dimension vector for token generation. Then we develop the Semantic Features of Words (SFW) loss for dialogue:

（8）

For convenience, the final SFW loss is denoted as the combination of the two parts:

（9）

Note that for clarity, are different views of the words’ semantic features. When pre-training the CRS models, only   contributes to the **recommendation module**, while   contributes to the **dialogue generation module**.

**3.2.2. Modelling semantic relevance between sentences**

In this section, we would like to model the semantic relevance between sentences. The main idea is to force the model to distinguish whether the **context–response pairs** are from the same dialogue with the advantage of contrastive learning framework. By this task, the fluency, coherence and consistency of generated utterances are improved.

1. Formally, given the **dialogue context utterance** 𝑋𝑢 with 𝑝𝑐 tokens and the **corresponding response utterance** with 𝑝𝑟 tokens, we treat them as positive pairs . Then we randomly sample 𝑛𝑔 negative responses  from the training corpus as the negative context–response pairs.
2. Next, we calculate the **context and response representations** as:  and  where Emb𝑐 is same as that in Eq. (7), and Emb𝑟 is the embedding layer for the response utterances.
3. Then we follow SimCLR and adopt the following **Semantic Relevance Between Sentence** (SRBS) loss to maximize the agreement of positive pairs  and minimize that of negative pairs , simultaneously:

一張含有 文字, 字型, 行, 白色 的圖片

自動產生的描述（10）

where 𝜏𝑠 is the **temperature hyper-parameter** for semantic knowledge, y− is the set of negative samples and sim(𝒛𝑥 , 𝒛𝑦 ) is the cosine similarity between 𝒛𝑥 and 𝒛𝑦 .

Now we develop two auxiliary self-supervision tasks to inject the semantic knowledge into the base CRS model.

4.Formally, the entire loss for semantic knowledge extraction is:

一張含有 文字, 字型, 白色, 行 的圖片

自動產生的描述(11)

where 𝑁𝑢 is the number of the samples belong to user 𝑢. To be noticed,  extracts the semantic knowledge for both the recommender and generation modules, and each part of  only contributes to the corresponding module.

3.3. Structural knowledge extraction

We then explore the intrinsic correlations of **user mentioned entities** on knowledge graphs for the recommender module. As the **entity embeddings** are learned by KGs, it contains useful structural knowledge, and plays a vital role in modelling the user preferences. For example, when the user said that he loves Tom Hanks, we could recommend ‘‘Cast Away’’ because it is starred by Tom Hanks according to DBpedia.

1.To learn the structural knowledge from the **entity embeddings**, we propose to recover the actual entity with the masked user mentioned entities. Given the 𝑗th sample of user mentioned entities  = {𝑒1 , …, 𝑒𝑘 , …, 𝑒𝑛 }, we randomly delete the 𝑘th entity and the rest entities   = {𝑒1 , …, 𝑒𝑘−1, 𝑒𝑘+1, …, 𝑒𝑛 } are leveraged to predict the masked entity 𝑒𝑘 as follows:

(12)

Here  is the **entity representation** of  is the entire entities’ representation matrix. Then the structural knowledge extraction loss is defined as:

一張含有 字型, 文字, 白色, 圖形 的圖片

自動產生的描述(13)

* 1. Cross knowledge extraction

The **entity embeddings** are learned from KGs with structural knowledge, which contains different types of knowledge compared to semantic words.

Thus we would like to fuse the knowledge from **semantic words** and **structural entities**. This is called **cross knowledge**, which considers the co-occurrence relations between words and entities. For example, when the user said that he loves the item entity ‘‘Star Wars’’, we may not understand his taste directly. But when he also mentioned words ‘‘fantasy’’, it is prone to say that he requires recommendations about science fiction films.

1. Thus, we would like to exploit the inter-information between semantic words and structural entities, and propose the following auxiliary training objective. Given the user mentioned entities set  and the corresponding word set , we view them as positive samples, and randomly collect 𝑛𝑔 negative word sets . Our goal is to minimize the difference between entities set and positive words, and maximize that between entities set and negative words. Therefore, contrastive learning framework is a good choice. For similarity function, we could simply leverage the cosine function. Nevertheless, we propose a novel **NLL-based similarity function**, which is very suitable for our task.
   * 1. NLL-based similarity function

As we concern on the co-occurrence relations between words and entities, the core idea of our function is: if the entity set could be correctly predicted by the word set, we assign the entity–word pair a high similarity score; otherwise, it would receive a low similarity score. To measure whether the mentioned entities are properly predicted, we adopt the NLL loss. Compared with the cosine similarity, this novel function has the advantages of distinguishing what kind of entities and words are prone to appear together.

1.Formally, the probability of entities 𝑒𝑘 is calculated by the word set as following

(14)user講的那個words是屬於哪個entity

where  denotes the **representations of the word set** helps to learn the probability of each entity. Then the **similarity function of entity and word set** is defined as:

(15)

where ℎ(𝑥) = 3 − 4 ∗ Sigmoid(𝑥∕𝜏𝑐1 ) is a normalized function to ensure that the value range of 𝑠𝑐 (⋅) is between −1 and 1. Finally, the cross knowledge loss is defined as

一張含有 文字, 字型, 筆跡, 行 的圖片

自動產生的描述(16)

where 𝜏𝑐2 is the temperature hyper-parameter for cross knowledge and 𝜆5 is the weight hyper-parameter of loss.（relevance between words and entities）

3.5. Model optimization

3.5.1. Item recommendation

The entire procedure of the recommender module consists of two stages, namely pre-training and fine-tuning stages.

At the first stage, the parameters of recommender module (including GCN, RGCN and self-attention parameters according to Eq. (2)) are optimized by combining three types of knowledge (Eq. (11), Eq. (13) and (16)):

一張含有 字型, 印刷術, 文字, 書法 的圖片

自動產生的描述(17)

At the fine-tuning stage, we initialize the recommender model by the learned parameters from the last stage, and train the entire recommendation parameters with the help of the **supervised signals**:

一張含有 文字, 字型, 白色, 行 的圖片

自動產生的描述(18)

3.5.2. Dialogue generation

Similar to the recommendation part, dialogue generation module also includes **two stages**.

1. At the **pre-training stage**, the self-supervised signal is leveraged to pre-train the embedding layers **Emb𝑐** , **Emb𝑟** and **transformation matrix 𝑊𝑐** (refer to Eq. (7)).
2. Then at the next stage, we adopt the following **knowledge-aware generator** to better incorporate the knowledge. Knowledge-aware Generator: When generating tokens at decoding phrase, we incorporate the knowledge representations S𝑢 and T𝑢 learned by the recommender part into decoder. Thus the generation part would benefit from the recommendation part. Concretely, we add the knowledge at the multi-head attention (Vaswani, et al., 2017) layer (MA):

一張含有 文字, 字型, 白色, 筆跡 的圖片

自動產生的描述(19)

S:word representation , T:entities representation, **utterance** 𝑋

where 𝑊\_𝑔𝑎𝑡𝑒 is gate function like Eq. (3) and FFN denotes the fully-connected feed-forward network. Then the fine-tuning objective of generation module is set as:

一張含有 文字, 字型, 白色, 行 的圖片

自動產生的描述(20)

生成對的response的機率