AtCRS: Attribute-based Conversational Recommender Systems

Anonymous ACL submission

Abstract

Conversational Recommender Systems are known to benefit from explanations of *why* an entity is recommended. In this work, we implement an attribute-based approach to generating such explanations, which we call AtCRS. We show that this approach is preferred by humans – making them feel more confident in the recommended item. We also show that attribute-first models provide benefits for automatic generation: AtCRS generates fewer hallucinations and is more consistent with the previous conversation than current state-of-the-art end-to-end systems. The newly annotated StrAtData and the code used in this paper will be made available upon acceptance.

1 Introduction

Previous work on Conversational Recommender Systems (CRS) has highlighted the importance for these systems to explain their recommendations (e.g. Tintarev and Masthoff, 2011; Nunes and Jannach, 2017). For example, there is evidence that explanations can help users to make better and faster decisions (Gedikli et al., 2014) and that they increase user trust (Pu and Chen, 2006). In a similar vein, our work explores whether recommendation strategies that contain an explanation in the form of a shared attribute will increase user trust/ confidence compared to conventional recommendation strategies.

In order to generate these recommendation strategies we explore current end-to-end models. Previous approaches to generating recommendations for CRS implement two separate processes: a recommendation module to predict preferred items and a conversation module to generate responses. Recently, Wang et al. (2022) proposed a unified method UniCRS using a prompt encoder to combine the entries from the external knowledge base with the history of the dialog. While this improves retrieval and word-overlap measures, it still leads to semantic inconsistencies. For example, their model often generates attributes which do not correspond to the <movie> placeholder, which is filled by the recommendation module later on, e.g. calling a recently released movie "a classic" as in Figure 1.

Context	
Seeker:	"Omg I love that movie Suicide Squad (2016)"
Recommender:	"I felt as if Suicide Squad (2016) was trying too hard and isn't as timeless as other dc movies such as **Man of Steel (2013)** or **The Dark Knight (2008)**"
Seeker:	"Is that the one with Christian Bale? I love him."
recOnly	"Try **Aquaman (2018)**"
acknRec	"Yeah same here. Have you seen **Aquaman (2018)** yet?"
recReason	"It's a really good movie! If you like The Dark Knight (2008) you would love **Batman Begins (2005)**. It has Christian Bale."
Hallucination:	"Have you seen the movie **Kaapa (2022)**? It's such a classic."

Figure 1: Annotated example of recommendation strategies from the Redial dataset between two humans playing the role of *Seeker* and *Recommender*.

To address this, we present a new attribute-based approach to CRS, which we call AtCRS. This approach leverages the use of Knowledge Graphs (KGs) and external knowledge bases to capture the relevant attributes that explain *why* a user likes an entity, and then uses this information as a supporting argument when generating recommendations. For example, if a user mentions that they like a movie because of its lead actor, the recommendation should highlight the shared attribute, see Figure 1.

2 CRS Recommendation Strategies

2.1 Task Definition

We define an attribute-based recommendation by topic shift when, given one or more preferences expressed by the user as O_1, the system recommends a product O_2 providing an explanation h on why it made that particular recommendation choice.

To implement this approach, we build upon previous work using topic-shifting via entity-bridging for open-domain dialogue by Sevegnani et al. (2021). Our experiments compare three strategies adapted from Sevegnani et al., see Figure 1:

- `recOnly`: No connection nor acknowledgment towards the seeker's previously expressed preferences.
- `acknRec`: The recommender first briefly acknowledges what the seeker previously said and provides a recommendation that is not necessarily connected to the seeker's preferences. There is no mention of a mid-way concept or NE.
- `recReason`: The recommender **repeats** a concept or a named entity (NE) from the seeker's last utterance to connect their preferences with the recommended product.

Our hypothesis is that the latter will increase user confidence in the recommended item.

2.2 Data

We apply these strategies in the context of CRS by extending the commonly used ReDial corpus (Li et al., 2018) for generating product recommendation strategies in the movie domain. We first filter the dataset to only include recommendation turns, `Redial-rec`. We classify a turn as "recommendation" if it contains a movie ID (defined with a tag beginning with "@") that has then been classified as "suggested" by both the *Seeker* and the *Recommender* in the original Redial data collection. Our results suggest that $\approx 20\%$ of all turns are recommendations. We further filter the recommendation corpus by keeping only turns that contain one or more movie IDs, or an attribute. We will call this sub-corpus `StrAtData`. We randomly sample 100 examples of `StrAtData` for manual annotation.

2.3 Human Preference

Next, we are interested which of the strategies are preferred by human raters. We randomly sampled 100 examples for each of the three recommendation strategies from the automatically annotated `StrAtData` sub-corpus. Crowd-workers were asked to evaluate the question "Are you confident that the response will enable the seeker to find the movie they are looking for?" on a Likert scale from 1 (Not at all) to 5 (Absolutely). We find that annotators show high agreement on this task ($\kappa = 87.3\%$). The results in Table 1 show that, as hypothesised, the `recReason` strategy obtains the highest rating. A Wilcoxon signed-rank test confirms that `recReason` is rated significantly higher than `acknRec` ($p < 0.01$) and `recOnly` ($p < 0.0003$).

	Rating
`recOnly`	3.4
`acknRec`	3.53
`recReason`	**3.81***

Table 1: Overall ratings for each recommendation strategy from the human evaluation. * indicates statistical significance $p \leq 0.01$ in regards to both the other strategies.

	RecReason	**AcknRec**	**RecOnly**
Manual Ann.	18%	13%	69%
Automatic Ann.	26%	4%	70%

Table 2: Manual and automatic annotation of the `StrAtData` corpus using the three strategies: *Recommend and Reason*, *Acknowledge and Recommend*, and *Recommend Only*.

3 Implementation of `AtCRS`

For implementing the new recommendation strategies, we build upon the prompt-encoder from `UniCRS` from Wang et al. (2022). In contrast to `UniCRS`, and in order to reduce inconsistencies, generation and item recommendations are not treated as two separate tasks. Whereas `UniCRS` first generates a template, which is then filled in by the system after movie selection, we first select the movie to recommend, and then we generate the recommendation sentence. During fine-tuning, we change the input encoding for both the prompt encoder and the DialoGPT LM, as shown in Appendix Figure 4: In addition to the inputs from the original `UniCRS` system, we provide the movie title(s) to be recommended, M_r, their corresponding genres G_m, and, finally, which strategy to use among the three identified in Section 2.2. We separate each of these inputs with sentinel tags: `[REC]` for the movie to recommend, `[GEN]` for the movie genres, and `[STRAT]` for the recommendation strategies.

	GT Similarity		*Content Consistency*			
	BLEU	**Dist@{2-4}**	**UserOverlap@{2-4}**	**Sent.**	**Acc.**	**Simil.**
`UniCRS`	**17.99**	0.85, **0.72**, 0.59	34.3, 20.2, 14.8	0.35	0.88	0.78
`AtCRS`	17.91	0.85/0.71/0.57	**34.9/21.0/16.0**	**0.37**	**0.97**	**0.92**
`AtCRS`-strategies	17.78	**0.86/0.72/0.6**	34.7/20.8/15.7	0.35	0.92	0.87

Table 3: Results comparing `UniCRS`, `AtCRS`, and `AtCRS`-strategies over the metrics described in Section 4.1.1.

Resulting in a model input of:

$$F_i=\{w_e||p_c||D_h||\texttt{[REC]}M_r||\texttt{[GEN]}G_m||\texttt{[STRAT]}s_t\} \quad (1)$$

where w_e is the fused word embeddings, p_c is the conversation-specific prompt, and D_h is the dialog history. The model is trained using cross-entropy loss L_{ce}:

$$L_{ce} = -\sum_{c=1}^{M} y_{o,c} log(p_{o,c}) \quad (2)$$

4 Experiments

4.1 Data, Baselines, and Metrics

We train and evaluate `AtCRS` to generate recommendation strategies, using the subset of the Redial corpus (`StrAtData`), as described in Section 2.2. We provide the previous dialog turns as 'gold' input context from the original dataset.

We compare the `AtCRS` approach with the original `UniCRS` model (Wang et al., 2022). Additionally, we show an ablation study between the original `AtCRS` model and the same model without encoding the strategy tag as input (`AtCRS`-strategies). Training details can be found in the Appendix.

4.1.1 Conversational Metrics

Automatically generated text is commonly evaluated measuring similarity with respect to a human written Ground Truth (*GT Similarity*), using standard metrics such as **BLEU** and Distinct n-gram score **Dist2-4**. However, these metrics do not account for important quality metrics in conversational search settings, such as coherence and consistency with the previous conversational context. As such, we propose a new set of metrics aimed at measuring *Content Consistency*:

UserOverlap@2-4 measures consistency with the user request (rather than the GT reference) using raw n-gram overlap between the predicted utterance and the last turn from the seeker.

Sentiment measures whether the recommendation matches the sentiment expressed by the user, aiming to distinguish likes and dislikes in user preferences. We set a sentiment threshold of 0.1, where if the difference between the two polarities is greater than the threshold, is considered as a non-match.

Strategy Accuracy determines whether the predicted recommendation strategy matches the one identified in the annotations.

Genre Similarity calculates BERTScore similarity between the predicted and previously mentioned genres, e.g. the user likes "horror" and the system recommends "a scary movie".

5 Results

The results in Table 3 show that `AtCRS` models exhibit superior performance compared to `UniCRS` across the majority of evaluation metrics, especially the ones which aim to measure *Content Consistency*. The biggest gains can be observed when predicting the type of recommendation strategy, as well as for measuring the similarity of the movie genre between user preference and recommendation.

The ablation study (`AtCRS`-strategies) supports the hypothesis that encoding the strategy information improves performance. This holds true for both genre similarity prediction and sentiment matching.

However, we also observe that the overall sentiment matching score is relatively low. Our analysis suggests that this may be due to the tendency of the `AtCRS` model to produce exaggerated utterances. For instance, the use of words such as "love" instead of "like" in the model outputs, although semantically similar, results in a higher polarity that exceeds the predefined threshold for sentiment matching.

Next, we investigate whether attribute-generation leads to fewer hallucinations. Related work by Maynez et al. (2020) further distinguishes these inconsistencies into "extrinsic hallucinations", where model generates facts that are not grounded in any source material, and "intrinsic hallucinations", where the generated output is "unfaithful" to the input, i.e. misrepresent information from the source. For our domain, an extrinsic

	Factual Hallucinations
UniCRS	0.262
AtCRS	**0.247**

Table 4: Comparison of factual hallucinations between AtCRS and UniCRS outputs.

Seeker:	"I need a movie to watch with my kids".
Recommender:	"I just recently watched Coco (2017) with my kids, they loved it. It is an animated movie but it has a really good plot and storyline and is really emotional at the end"
Seeker:	"Yes is very good i watch that movie every day"
UniCRS *Output:*	"That is a great movie."
AtCRS Output:	"Have you seen Home Alone (1990)? That is another good one. My son is 10 and loves it."
Reference:	"I also like several classics such as Home Alone (1990) and the sequels. Do you prefer more current family films?"
Seeker:	"Hi can you help me. I am looking for a good horror flick"
UniCRS *Output:*	"Sure."
AtCRS *Output:*	"I love horror films! I have seen Scream 2 (1997) and enjoyed it."
Reference:	"I like the whole serious Scream 2 (1997) was another good one"

Figure 2: Examples of RecReason strategies from the AtCRS model outputs.

factual hallucination corresponds to all concepts outside of the movie genre domain, i.e., not in the MovieLens genre list. Intrinsic faithfulness hallucinations correspond to substitutions. For example, the predicted genre 'drama' against the reference 'comedy' is an example of a faithfulness hallucination. We measure intrinsic faithfulness hallucinations using BERTScore similarity. The results in table 4 show that AtCRS generates less factual hallucinations than UniCRS and, as proven from the Similarity results in Table 3, there are fewer actual faithfulness hallucinations.

In order to further illustrate our results, we provide a short **example-based analysis**. The first example in Table 2 demonstrates a scenario where the seeker expresses the attribute *kids* as a preference, which is picked up by AtCRS and *"my son is 10"* is used as an attribute-based explanation for the recReason strategy. While this strategy diverges from the GT recOnly, our results in Section 2.3 show that humans significantly prefer recReason over recOnly. In contrast, the UniCRS only acknowledges the movie preference.

In the second example in Table 2 demonstrates a similar point: Here, the seeker specifies the attribute *horror* as a preference. Despite the high similarity in surface form between the GT reference and the AtCRS output, the strategies employed are different. AtCRS utilises a recReason strategy that explicitly mentions the attribute *horror*, while the GT implements a recOnly strategy.

6 Conclusions

We present a new attribute-based approach to generating recommendation strategies for Conversational Recommender systems. We first annotate three different conversational recommender strategies, extending previous work. We then show that humans prefer strategies that contain an attribute-based explanation (recReason). We then implement these strategies by extending the state-of-the-art UniCRS system to be attribute-driven, where we encode contextual information in the input.

Our results show that our attribute-based system AtCRS outperforms UniCRS on the majority of metrics, but most notably on metrics measuring content consistency, which indicates that AtCRS not only leads to strategies which are similar to the human-generated ground-truth, but also to strategies which are more convincing by providing an explanation of *why* an entity was recommended.

The current method of evaluation requires improvement, as the standard metrics employed in this study were not comprehensive enough to fully gauge the performance of the model. The limitations of using reference outputs to compare the model predictions have been highlighted, as the references themselves contain noise such as spelling mistakes and extraneous information. To overcome this, an alternative method of evaluating the model was proposed by comparing the model's predictions with the context instead of the reference, however, this method was also not deemed to be fully effective.

In conclusion, it is important to acknowledge the potential risks associated with the use of conversational models. One concern is that such a model could be trained in a malevolent manner, leading to malicious behaviour that could result in the manipulation of consumer behaviour and raise ethical concerns. In conclusion, while conversational models have the potential to improve human-computer interactions, it is important to consider the potential risks and unintended consequences of such technology.

References

Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. 2014. How should i explain? a comparison of different explanation types for recommender systems. *Int. J. Hum.-Comput. Stud.*, 72(4):367–382.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. *Advances in neural information processing systems*, 31.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Ingrid Nunes and Dietmar Jannach. 2017. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27(3–5):393–444.

Pearl Pu and Li Chen. 2006. Trust building with explanation interfaces. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, IUI '06, page 93–100, New York, NY, USA. Association for Computing Machinery.

Karin Sevegnani, David M Howcroft, Ioannis Konstas, and Verena Rieser. 2021. Otters: One-turn topic transitions for open-domain dialogue. *arXiv preprint arXiv:2105.13710*.

Nava Tintarev and Judith Masthoff. 2011. *Designing and Evaluating Explanations for Recommender Systems*, pages 479–510. Springer US, Boston, MA.

Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1929–1937.

A Appendix

	UniCRS	`AtCRS`
Tokenizer	DialoGPT-small	DialoGPT-medium
Language model	DialoGPT-small	DialoGPT-medium
Text tokenizer	Roberta-base	Roberta-base
Text encoder	Roberta-base	Roberta-base
Gradient accumulation steps	1	1
Train batch size	8	4
Eval batch size	16	8
Num warmup steps	6345	772
Context max length	200	512
Response max length	183	185
Num train epochs	10	15

Table 5: Training details for the conversation task for both `UniCRS` and `AtCRS`

B Human Evaluation

In order to understand whether users prefer certain strategies based on the turn position inside the dialog, we picked the same percentage of examples (33%) from one of the three dialog positions: beginning, middle, end. Given that the average dialog length is 13 turns, we consider the beginning as turn 1-4, middle as turn 5-10, and end of the dialog as turn 11-13 (and beyond). In order to evaluate confidence/trust, we asked crowd-workers to evaluate the question"Are you confident that the response will enable the seeker to find the movie they are looking for?" on a Likert scale from 1 (Not at all) to 5 (Absolutely). In order to assure annotation quality we selected Master workers, located in an English-speaking country, and had submitted at least 1000 tasks with an acceptance rate of at least 90%. These constraints help guarantee the quality of the collected data, excluding users clicking on

	Train	**Dev**	**Test**
`Redial`	163.8k	18.5k	23.9k
`Redial-rec`	33.9k	3.8k	4.4k
`StrAtData`	25.9k	2.8k	3.5k

Table 6: Difference in size between the `train-dev-test` split of all three datasets mentioned in Section 2.2. `Redial` is the original dataset, `Redial-rec` corresponds to a subset of the first containing the recommendation utterances only, and `StrAtData` is a further subset including at least one movie or concept mention.

the task just for its payment.

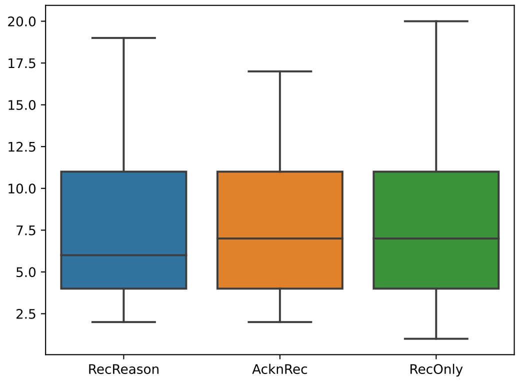

Figure 3: Preference for turn position in the dialog for each of the recommendation strategies.

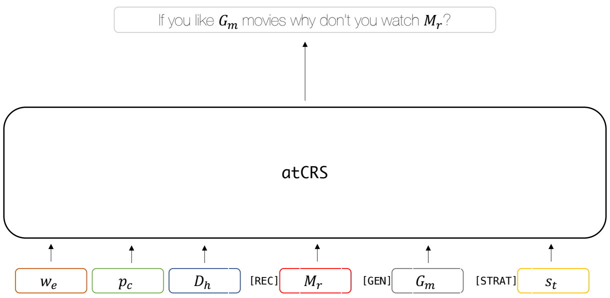

Figure 4: Revisited input for the AtCRS model.

	Rating
same strategy, low BLEU	3.69
high BLEU, diff strategy	3.82

Table 7: Human evaluationof the AtCRS model outputs. Results are not statistically significant.

Seeker:	"Hi! I am looking for a romantic comedy like My Best Friend's Wedding (1997)".
Recommender:	"Hey there, **Pretty Woman (1990)** was a good one."
Seeker:	"I've seen it! But I like all of the Julia Roberts movies."
Recommender:	"Then if you like Julia Roberts you got a classic like **Eat Pray Love (2010)**."
Seeker:	"Hey any good recommendations for Disney movies? I like the musical ones."
Recommender:	"Have you seen **Coco (2017)** or **Trolls (2016)**? **High School Musical** is another good one too."
Seeker:	"I love love love the High School Musical. My all time favorites are Beauty and the Beast (1992), A Goofy Movie (1995) , The Hunchback of Notre Dame (1996) and Snow White and the Seven Dwarfs (1937) just to name a few."
Recommender:	"Did you know that **Captain America** is also a Disney movie?"

Figure 5: Examples of a RecReason strategy for recommendation.

Recommender:	"What are your 3 favorite genres?"
Seeker:	"I like comedy and action".
	. . .
UniCRS *Output:*	"Yes, they are pretty good. Have you seen any movies recently?"
AtCRS *Output:*	"Tropic Thunder (2008) is a funny comedy. It is a little bit dark."
Reference:	"I know Kevin Hart had a part in it, too. Have you seen Tropic Thunder (2008)?"
Recommender:	"How are you today?"
Seeker:	"Hi, I liked The Purge (2013) can you recommend something similar?
Recommender:	What kind of movies do you like? my favorite is Fight Club (1999)
Seeker:	"I haven't seen it. How about a war movie then?"
UniCRS *Output:*	"Maybe a action movie like The Shawshank Redemption (1994), or a full on drama like The Terminator (1984) or a horror movie like Kiss the Girls (1995)"
AtCRS *Output:*	"There's also Enemy at the Gates (2001). It's good and violent, but well deserved."
Reference:	"Enemy at the Gates (2001) is really good snipers during the battle of Stalingrad"

Figure 6: More examples of model outputs