DSC 261 Project Final Report: Counterfactual Explainable Recommendation using LLMs

Aman Parikh | Kartikay Anand
HDSI
UC San Diego La Jolla, CA 92093
amparikh@ucsd.edu | kaanand@ucsd.edu

Abstract

Recommendation systems have become an integral part of various online platforms, enhancing user experience by providing personalized content. However, there is a growing need for more transparent and interpret able recommendation systems to address user trust issues. This project focuses on counterfactual reasoning for explainable recommendation. The design of evaluation metrics is from two viewpoints: 1) user's perspective and 2) model's perspective. In this project we will experiment with different Large Language Models (LLMs) both open source and proprietary(till no cost limit) like GPT3.5, Llama2, Mistral to extract aspects and sentiments, and further compare it with the approach used by the paper (Sentires). Code can be found here

1 Introduction

Problem Definition: Counterfactual reasoning seeks to comprehend the fundamental mechanism that genuinely prompted the recommendation by implementing interventions on various aspects and observing the outcomes. Basically we alter a specific aspect which is important to the user on the item's side and see if the there is any difference in the ranking of the the item on the recommendation list of the user

Problem Significance: This is significant because our approach produces explanations from a counterfactual standpoint: if an item would not have received a recommendation with a slight decrement in performance on certain aspects, then these specific aspects become the rationale for the model recommending the item. This provides insights to recommender system designers as well increase trust of the users in the model.

Technical Challenge: Current approaches neglect the complexity and strength of explanations, lacking the capability to determine the optimal number of aspects to employ during explanation generation. While many methods generate explanations based on a single aspect, the actual reason behind a recommendation may be influenced by multiple aspects. We will use LLMs instead of original paper's library(Sentires) to see the affect of aspect generation using LLMs on metrics. LLMs can infer implicit sentiment and overall tone from the text as they are trained on huge data and updated regularly.

State-of-the-Art:

EX^3: Explainable Attribute-aware Item-set Recommendations[3]: EX3 enhances recommendation systems by autonomously identifying significant user-preferred attributes and suggesting item sets. It eliminates the need for manual annotations, offering users more transparent and personalized

Preprint. Under review.

purchase guidance. The Extract-Expect-Explain approach ensures a multi-step adaptive learning process, contributing to the state-of-the-art in recommendation technology.

Limitations: The limitation of these recommendation approaches lies in their reliance on a common hypothesis: identifying aspects that best align with the user's preference and the item's performance as the explanation for a recommendation. This might overlook certain things(like nuanced changes), that a counterfactual approach can account for.

Contributions: We tried to harness the power of LLMs to generate Aspect, Opinion and Sentiment triplets and compare the performance difference on metrics with the original paper.

2 Related Work

"Counterfactual Explainable Recommendation" by Tan et al. (2023)[2]: The paper introduces CountER (Counterfactual Explainable Recommendation), a novel approach to explainable recommendation systems that leverages counterfactual reasoning to provide clear explanations for users and system designers. CountER formulates a joint optimization problem for recommended items, generating counterfactual items with minimal changes to explain why the original item was recommended, while utilizing the Sentires to extract aspects, opinions, and sentiment triplets. It proposes evaluation metrics from both user and model perspectives, demonstrating superior performance on five real-world datasets compared to existing models, with its source code available for further exploration and adoption.

"Generative approach to Aspect Based Sentiment Analysis with GPT Language Models" by Chumakov et al. (2023)[1]: The paper introduces an innovative approach to Aspect Sentiment Triplet Extraction (ASTE) using open-domain generative methods based on Generative Pre-trained Transformers (GPT), offering a solution to the limitations of existing ASTE techniques reliant on BERT embeddings and large manually-tagged datasets. By employing few-shot and fine-tuning strategies, the proposed method demonstrates effectiveness in extracting representative features from textual data, structuring output triplets consistently, simplifying terms without loss of meaning, and analyzing data from unknown domains.

Technical Differences from our proposed method: In contrast to the above approaches, our method explores LLMs with a few shot as well as 0 shot approaches to generate aspects, opinions and sentiments on the data. The LLMs sucessfully generated the triplets for almost every textual reviews whereas the Sentires success rate was very low. We combine both aspects from the papers and analyse the results.

3 Methodology

Problem Setting: Let binary matrix B be the user-item interaction matrix, where $B_{i,j}$ = 1 if user u_i interacted with item v_j ; otherwise, $B_{i,j}$ = 0. R(u,K) is used to represent the top-K recommendation list for a user u, and it is said that $v \in R(u,K)$ if item v is recommended to user u in the user's top-K list. We will use the few shot approach of LLMs to extract (Aspect, Opinion, Sentiment) triplets from the textual reviews. Besides, suppose we have a total number of r item aspects $A = a_1, a_2, \cdots, a_r$,then the user-aspect preference matrix X and the item-aspect quality matrix Y are computed. $X_{i,k}$ indicates to what extent the user u_i cares about the item aspect a_k. Similarly, $Y_{j,k}$ indicates how well the item v_j performs on the aspect a_k. More specifically, X and Y are calculated as:

$$X_{i,k} = \begin{cases} 0, \text{ if user } u_i \text{ did not mention aspect } a_k \\ 1 + (N-1)\left(\frac{2}{1+\exp(-t_{i,k})} - 1\right), \text{ else} \end{cases}$$

$$Y_{j,k} = \begin{cases} 0, \text{ if item } v_j \text{ is not reviewed on aspect } a_k \\ 1 + \frac{N-1}{1+\exp(-t_{j,k} \cdot s_{j,k})}, \text{ else} \end{cases}$$

Figure 1: Equation for user-aspect and item-aspect preference matrices

where N is the rating scale in the system, $t_{i,k}$ is the frequency that user u_i mentioned aspect a_k. $t_{j,k}$ is the frequency that item v_j is mentioned on aspect a_k , while $s_{j,k}$ is the average sentiment of these mentions.

Idea Summary: A black-box recommendation model is used that predicts the user-item ranking score $s_{i,j}$ for user u_i and item v_j by: $s_{i,j} = f(X_i, Y_j \mid Z, \theta)$; where θ is the model parameter and Z could be ratings, clicks etc. The network concatenates the user's and the item's aspect vectors as input and outputs a one-dimensional ranking score $s_{i,j}$. Then, the model is trained with a cross-entropy loss. The basic idea is to extract triplets using few shot prompts from LLMs. Some problems include extracting too many and too specific aspects from the data which can make it difficult to associate with a user and item. These can be taken care of by adjusting the temperature of the LLMs.

Description: Suppose item v_j is in the top-K recommendation list for user u_i. As mentioned before, the counterfactual reasoning model aims to find simple and effective explanations for v_j , which can be shown as the following constrained optimization framework, minimize Explanation Complexity (C) s.t., Explanation is Strong (S) enough. Mathematically, the framework can be realized with the following specific optimization problem,

$$\begin{aligned} \text{minimize } & C(\Delta) = \|\Delta\|_2 + \gamma\|\Delta\|_0 \\ \text{s.t., } & S(\Delta) = s_{i,j} - s_{i,j_\Delta} \geq \epsilon \end{aligned}$$

Figure 2: Equations for minimizing Complexity while Strength of the Explanation is above a threshold

where $||\Delta||_0$ are non-zero values in the aspect vector and $||\Delta||_2$ is how many changes need to be applied on these aspects, which can be represented as the sum of square of Δ and γ is a hyperparameter. Here, $s_{i,j}$ is the original ranking score of item v_j , and s_{i,j_Δ} is the ranking score of v_j after Δ is applied to its quality vector. Here, $\in$ is threshold which is the margin between item v_j's score and the K+1's item's score in the original recommendation list.

Implementation: We used the codebase mentioned in the paper for the counterfactual part and had to preprocess the data differently due to difference in the extraction process of LLMs compared to Sentires. We also experimented with different learning rates like 0.001, 0.005, 0.01, 0.05. The model performed well and converged for learning rate= 0.01. For the LLM part we had to go through the documentation of every LLM and design the prompt as well adjust the parameters like temperature so that the output is in the desired format as well as is coherent. We also adjusted the temperature values to see the difference in aspect generation. Higher temperature gave distinct responses but the responses were limited due to the distinctness of the output. So we set temperature as low as aspects can be repeated across reviews.

4 Experiments

Datasets and Tools:

4.1 Dataset

We used core datasets which contain users with a minimum no of textual reviews. The Amazon datasets we explored are as follows :

1. Amazon Cellphones and Accessories Dataset We have done some preprocessing in order to shorten the dataset. Firstly, we make it a 12 core and have filtered out reviews with a legnth of 500. The following shows the number of users, items and interaction in our filtered dataset.
   (a) Items: 4397
   (b) Users: 557
   (c) Reviews: 9011

2. Clothing, Shoes and Jewelry Dataset (18-core) Here, we have filtered 18 core data. The review length filtering is the same.
   (a) Users: 411
   (b) Items: 7043
   (c) Reviews: 10268

4.2 Tools

We plan to use the following tools : (1) Python (2) Pytorch (3) Google Collab (4) Jupyter notebook (5) Hugging Face (6) Sentires

4.3 Proposed Baseline

The baseline we are using is using the Sentires method. The quantitative values of the baseline can be found in the quantitative evaluation section.

4.4 Large Language models

We experimented with a large variety of language models, both open source and proprietary. Here is a full list of languge models we explored.

(a) GPT3.5 Turbo Instruct
(b) Mistral(7b and 8X7B)
(c) LLAMA2 7B and ABSA Fine Tuned LLAMA2 7B.
(d) Phi2
(e) PaLM

We have discussed and shared the outputs of some them below:

(a) **LLama 2(7B)** We implemented the LLama2 7B variant of the LLama series. Below is the output we got: As can be seen above, while the models does well to extract

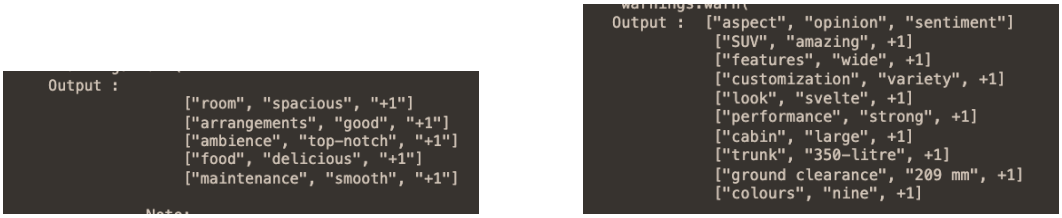


Figure 3: LLama2(7B) Triplet Extraction

the aspects, opinion ad sentiment triplet, the formatting of he output is not consistent, which makes it difficult for us to extract the output and preprocess.

(b) mistral(8x7B) We also tried the Mistral 8x7B as well as the Mistral 7B models. Both of these were quantized to 4 bit. The mistral 8x7B model produced good output where the 7B model was not able to generate the output accurately. The following image shows the output generate by the mistral 8x7b model.

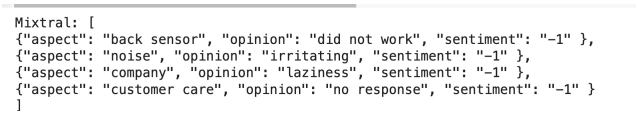


Figure 4: Mistral 8x7B Output

As can be seen the outputs produced by the model are good in quality, but the model is computationally expensive, as a result of which it was not possible to apply to an entire dataset.

(c) Fine Tuned LLama2 7B model We also used the fine-tuned version of the Llama-2-7b model, optimized for Aspect-Based Sentiment Analysis (ABSA) using a dataset of 2000 sentences. The fine tuning helps the model accurately capture aspects, opinions and related sentiments from sentences. The following image shows the output from the fine tuned model:

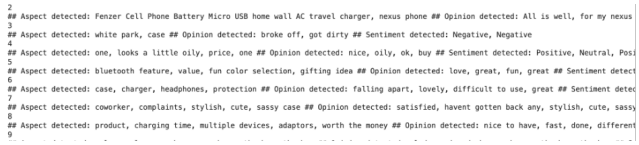


Figure 5: Fine Tuned LLama 2 Output

As can be seen above, the fine tuned model was able to extract good quality triplets from the review with consistent output format which allowed us to use the model for the entire dataset. The computationally expensive nature of the model did not allow for the model to run on the entire dataset and was able to process around 600 rows, before the kernel timed out.

(d) GPT3.5 Turbo Instruct This is the model we finally used to create the dataset after exploring other models. While being a proprietary model, GPT provides initial credit of 5 dollars and thus we were able to effectively use it without incurring high expenses. The following is the output of the GPT3.5 model:

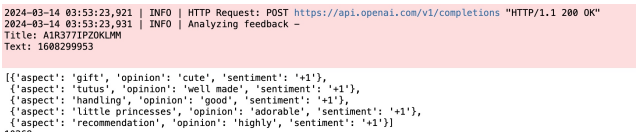


Figure 6: GPT 3.5 Output

As can be seen in the above figure, GPT 3.5 provided good quality outputs and was not too much computationally expensive. As a result, the triplets were obtained from this model for a dataset of 10000 rows.

4.5 Evaluation

Quantitatively evaluating explanations is crucial. We use two types of evaluation metrics in this section: user-oriented evaluation and model-oriented evaluation.

User-oriented Evaluation

In user-oriented evaluation, we rely on users' reviews to determine the ground-truth reasons for purchasing an item. Using the textual reviews, we extract positive sentiment aspects ($P_{i,j}$) mentioned by the user u_i for an item v_j. Our model generates an explanation vector (Δ), and the aspects corresponding to non-zero values in Δ constitute the explanation.

For each user-item pair, precision, recall, and $F1$ score are calculated, comparing the generated explanation Δ with the ground-truth vector $P_{i,j}$.

Model-oriented Evaluation

User-oriented evaluation checks the consistency of generated explanations with user preferences but does not assess whether the explanation justifies the model's behavior. To address this, we propose two scores for model-oriented evaluation: Probability of Necessity (PN) and Probability of Sufficiency (PS).

Probability of Necessity (PN): PN evaluates the extent to which a condition is necessary. For each user-item pair, we alter the aspect values in the item-aspect quality matrix Y based on the generated explanation. PN is calculated by assessing the likelihood that, if the explained aspects did not exist, the item would not be recommended.

$$\text{PN} = \frac{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in R_{i,K}} \text{PN}_{ij}}{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in R_{i,K}} I(\mathcal{A}_{ij} \neq \emptyset)}, \text{ where } \text{PN}_{ij} = \begin{cases} 1, & \text{if } v_j^* \notin R_{i,K}^* \\ 0, & \text{else} \end{cases}$$

Figure 7: Equation for Probability of Necessity (PN)

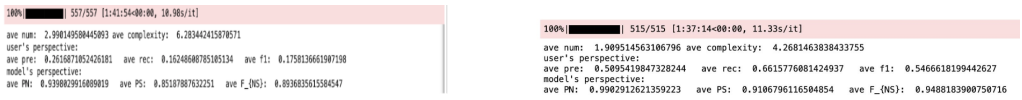
Probability of Sufficiency (PS): PS evaluates the extent to which a condition is sufficient. We alter aspect values in Y based on the generated explanation, considering a counterfactual world where only the explained aspects exist. PS is calculated by assessing the likelihood that, if only the explained aspects existed, the item would still be recommended. Finally,

$$\text{PS} = \frac{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in R_{i,K}} \text{PS}_{ij}}{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in R_{i,K}} I(\mathcal{A}_{ij} \neq \emptyset)}, \text{ where } \text{PS}_{ij} = \begin{cases} 1, & \text{if } v_j' \in R_{i,K}' \\ 0, & \text{else} \end{cases}$$

Figure 8: Equation for Probability of Sufficiency (PS)

we calculate the harmonic mean of PS and PN ($F_{NS} = \frac{2 \cdot PN \cdot PS}{PN + PS}$) to measure overall performance in model-oriented evaluation.

4.6 Quantitative Results



(a) GPT 3.5 Output

(b) Sentires Output

Figure 9: Amazon Cell Phone and Accessories Dataset

The results clearly indicate that using Sentires is clearly a better choice in our experiment. The average complexity is clearly much lower for Sentires than it is for the GPT 3.5 output. Similarly, the user oriented evaluation metrics are much higher for Sentires than for GPT.The model side orientation is comparable for both the techniques and infact outperformed the sentires method for the Amazon Fashion dataset.

The shortcoming of GPT in terms of low user side metrics and high complexity might be occurring due to the number of aspects. GPT is much more accurate than Sentires and captures much more aspects.Due to these vast number of aspects from GPT, it can becomes difficult to identify the most relevant aspects for a particular user. This can lead to inaccurate or irrelevant recommendations. Also, there might not be enough user data associated with each one to make accurate recommendations.

We also tried training the Base recommendation models with different batch sizes and observed the differences in metric performances.Batch size of 64 gave a better convergence compared to batch size of 128 and 256.

4.7 Qualitative Evaluation

The image13 provided a good idea of the quality of aspects generated by each method. It is clear that sentires only generates 1 aspects, while GPT 3.5 generated multiple high quality aspects for the same review. This is in accordance to what we discussed in the quantative evaluation section.

As can be clearly seen here14 as well, GPT is not able to capture aspects from only 663 out of the 9011 while on the other hand Sentires is not able to capture aspects from 6808 out of 9011 reviews. This results in a huge gap in the number of aspects created by both and thus we see the difference is evaluation.

We also explored zero shot learning and few shot learning. As we see here16 Zero-shot prompt did not work in generating the necessary output, so we used few-shot technique for the project.

(a) GPT 3.5 (b) Sentires

Figure 10: Amazon Clothing Dataset

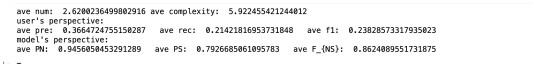
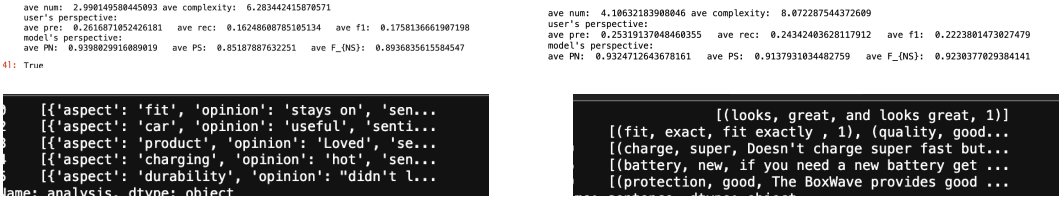


Figure 11: Effect of Batch Size (64,128,256) on Cell Phone Data

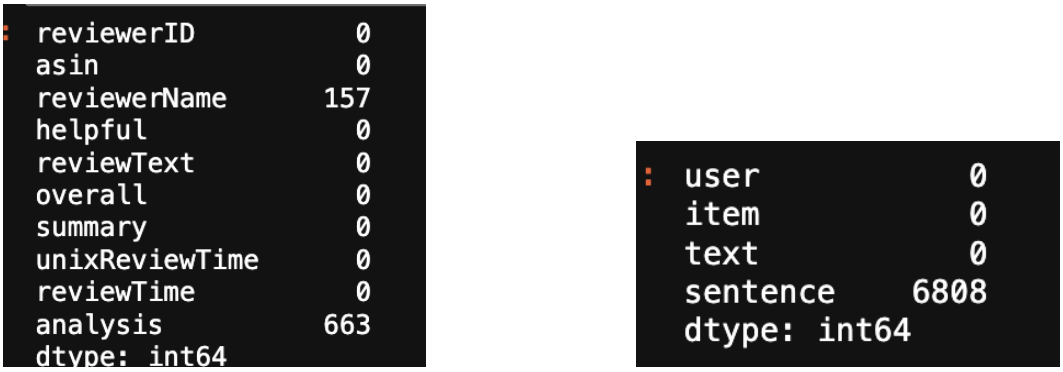


(a) GPT 3.5 (b) Sentires

Figure 12: Aspects Generated

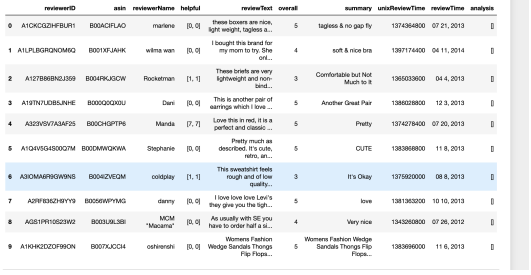
(a) GPT 3.5 (b) Sentires

Figure 13: Aspect Review Comparison



(a) GPT 3.5 (b) Sentires

Figure 14: Comparison of GPT 3.5 and Sentires for number of reviews processed



	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime	analysis
0	A1CKCGZIHFBUR1	B00ACIFLAO	marlene	[0, 0]	these boxers are nice, light weight, tagless a...	5	tagless & no gap fly	1374364800	07 21, 2013	[]
1	A1LPLBGRQNOM6Q	B001XFJAHK	wilma wan	[0, 0]	I bought this brand for my mom to try. She ont...	4	soft & nice bra	1397174400	04 11, 2014	[]
2	A127B86BN2JJ59	B004RKJGCW	Rocketman	[1, 1]	These briefs are very lightweight and non-bind...	3	Comfortable but Not Much to It	1365033600	04 4, 2013	[]
3	A19TN7UDBSJNHE	B000QGDX0U	Dani	[0, 0]	This is another pair of earrings which I love ...	5	Another Great Pair	1386028800	12 3, 2013	[]
4	A323V5V7A3AF25	B00CHGPTP6	Manda	[7, 7]	Love this in red, it is a perfect and classic ...	5	Pretty	1374278400	07 20, 2013	[]
5	A1Q4V5G4S00Q7M	B00DMWQKWA	Stephanie	[0, 0]	Pretty much as described. It's cute, retro, an...	5	CUTE	1383868800	11 8, 2013	[]
6	A3IOMA6R9GW9NS	B0042VEQM	coldplay	[1, 1]	This sweatshirt feels rough and of low quality...	3	It's Okay	1375920000	08 8, 2013	[]
7	A2RIB36ZHIIYY9	B00S6WPYMG	danny	[0, 0]	I love love love Levi's they give you the figh...	5	love	1381363200	10 10, 2013	[]
8	AGS1PR10S23W2	B003L9L3BI	MCM "Macarina"	[0, 0]	As usually with SE you have to order half a si...	4	Very nice	1343260800	07 26, 2012	[]
9	A1KHK2DZOF99ON	B007XJCCI4	oshirenshi	[0, 0]	Womens Fashion Wedge Sandals Thongs Flip Flops...	5	Womens Fashion Wedge Sandals Thongs Flip Flops...	1383696000	11 6, 2013	[]

Figure 15: No output in Zero-shot prompt in GPT3.5

Figure 16: Zero-shot prompt

5 Conclusion and Discussion

We looked at the effect of LLMs in aspect generation based sentiment analysis of textual reviews and the level to which it affected the user side metrics(Precision, Accuracy, F1) and model side metrics (PN, PS, FNS). Though Sentires performed better on both parameters than LLMs, that could be due to the long context window and thus large number of aspects generated by the LLMs. We learned a lot about counterfactual explanations as well metrics like PS, PN scores. We also learnt a lot about the hyperparameters of LLMs. In the future, given more resources we would like to fine tune LLM models to generate more precise aspect and sentiments. While, we tried fine tuning a small language model namely PHI2 (2b) parameter model ,we were limited by GPU resource timed out. We believe fine tuning for this task would be immensely helpful. In general, we found that for tasks like aspect based sentiment analysis overall, large large models have a lot of things to offer.

References

[1] Stanislav Chumakov, Anton Kovantsev, and Anatoliy Surikov. Generative approach to aspect based sentiment analysis with gpt language models. *YSC*, 2023.

[2] Juntao Tan, Shuyuan Xun, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. Counterfactual explainable recommendatio. *arXiv*, 2023.

[3] Yikun Xian, Tong Zhao, Jin Li, Jim Chan, Andrey Kan, Jun Ma, Xin Luna Dong, Christos Faloutsos, George Karypis, S. Muthukrishnan, and Yongfeng Zhang. Ex3: Explainable attribute-aware item-set recommendations. *ACM*, 2021.