Generating and Validating Contextually Relevant Justifications for Conversational Recommendation

Sergey Volokhin
Emory University
Atlanta, Georgia, USA
svolokh@emory.edu

Marcus D. Collins
Amazon
Seattle, WA, USA
collmr@amazon.com

Oleg Rokhlenko
Amazon
Seattle, WA, USA
olegro@amazon.com

Eugene Agichtein
Amazon
Seattle, WA, USA
Emory University
Atlanta, Georgia, USA
eugeneag@amazon.com

ABSTRACT

Providing a justification or explanation for a recommendation has been shown to improve the users' experience with recommender systems, in particular by increasing confidence in the recommendations. However, in order to be effective in a conversational setting, the justifications have to be appropriate for the conversation so far. Previous approaches rely on a user history of reviews and ratings of related items to personalize the recommendation, but this information is not generally available when conversing with a new user, and as such a cold-start problem imposes a challenge in generating suitable justifications. To address this problem, we propose and validate a new method, CONJURE (CONversational JUstificatons for REcommendations) to generate contextually relevant justifications for conversational recommendations. Specifically, we investigate whether the conversation itself can be used effectively to model the user, identify relevant review content from other users, and generate a justification that boosts the user's confidence in and understanding of the recommendation. To implement CONJURE, we test several novel extensions to prior algorithms, by exploiting an auxiliary corpus of movie reviews to construct the justifications from extracted pieces of those reviews. In particular, we explore different conversation representations and ranking approaches. To evaluate CONJURE, we developed a pairwise crowd task to compare justifications. Our results show large, significant improvements in Efficiency and Transparency metrics over the previous non-contextualized template-based methods. We plan to release our code and an augmented conversation corpus on Github.

KEYWORDS

conversational recommendations, explainable recommendations

ACM Reference Format:
Sergey Volokhin, Marcus D. Collins, Oleg Rokhlenko, and Eugene Agichtein. 2022. Generating and Validating Contextually Relevant Justifications for Conversational Recommendation. In *Proceedings of the 2022 ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR '22), March 14–18, 2022, Regensburg, Germany.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3498366.3505789

1 INTRODUCTION

Recommender systems' users benefit from understanding why or how a system came up with its recommendations [6, 9, 20, 25]. User-generated content, such as product or movie reviews, or social media posts, helps users to express their experiences and interests. Recommender systems have successfully used that content to infer preferences and improve recommendations. Such user-generated content could also help recommender systems generate finer-grained and more reliable explanations, in turn helping users make easier, more informed decisions [26].

In this paper, we address three challenges. First, we seek to generate justifications that can be used in conversations. Second, many customers have no history of reviews or other content from which to infer preferences or to find similar users' reviews from which justifications could be generated. Last, we wish to have a system that will not introduce factual errors into justifications. In the present work, we empirically investigate several proposed methods to generate justifications from review text, adapting them to a conversational setting. Moreover, we attempt to substitute the conversation itself for a user history or reviews.

Specifically, our contributions are:

(1) We investigate whether we can successfully use a conversation in place of a profile of the current user, to generate recommendation justifications.
(2) We comprehensively investigate how to generate movie recommendation justifications, using external movie reviews.
(3) We analyze the conditions under which the justifications are perceived to have better quality.

The rest of this paper is structured as follows: §2 describes recent work to generate justifications, especially from reviews. §3 describes our baseline systems and proposed new methods, particularly the relevance of an external item review to the conversation

CHIIR '22, March 14–18, 2022, Regensburg, Germany

ACM ISBN 978-1-4503-9186-3/22/03…$15.00
https://doi.org/10.1145/3498366.3505789

between a user and a recommender[1]. Section 4 describes the augmented dataset, initially constructed from movie reviews from the OpenDialKG corpus [13]. We then describe our crowd-sourced annotation efforts to evaluate the transparency and effectiveness of the generated justifications. We report and discuss the results of those experiments, and suggest future directions, in §5.

2 RELATED WORK

There has been a considerable recent effort to make use of user-generated review text to provide explanations or justifications for recommender systems, including Matrix[27] and Tensor[4] Factorization, Aspect Extraction [14, 15], Topic Modeling, and Word Clouds[12, 24], and Graph-based methods [8]. Wu et al. in [23] propose a Deep Conversational Critiquing Framework that provides explanations based on *contextualized descriptive key-phrases* together with the recommended item. The key-phrases consist of uni- and bi-grams, which are extracted beforehand from reviews using a custom algorithm. While they do achieve high performance of up to 88% of relevant key-phrases returned, those key-phrases are not formed into a coherent sentence that can be used in a conversational setting, for example, returned by a voice assistant or chat-bot. Here, we aim to generate justifications that can be used naturally in conversation.

BERT probing was tried for Conversational Recommendations [17], and while the authors show their improvement for recommendations, they also show that the accuracy of probing is rather low, with correct genres appearing in top-5 predictions for only 30-50% of cases. In our case, however, justifications need to be factually correct, so this technique can not be used for generating justifications.

Justification systems typically require that the system user has a pre-existing history of product interactions, reviews, or even specifically justifications, like [5, 8, 15, 23]. These systems learn from each user's review text any latent "aspects" in which the user is interested, but this does not apply to the majority of users, who in general have written few if any reviews. Moreover, these methods typically do not consider a current conversation, which may contain additional preferences. Thus, if a user has expressed interest in a particular director or actor in the past review, these systems could make use of that in justifying a new recommendation, but that same interest expressed in a new conversation would be missed. Some of the systems described above directly use review text to form the justifications (*e.g.*, [14]), though often not in a conversational way [8, 23]. Others use deep neural network models to generate text or fill in positive-sentiment templates (*e.g.*, [5, 15]) which may lead to inaccuracies.

3 METHODS

In this section we list the baselines we have used to evaluate our methods, and provide a detailed overview of our proposed method of CONversational JUstificatons for REcommendations (CONJURE).

Table 1: Categories and subcategories of templates with their associated probability

Category	Subcategory	Example
Movie Features	Cast (7.2%)	This movie features %MAIN ACTOR%.
Movie Features	Director (7.2%)	It is directed by %DIRECTOR%.
Movie Features	Genre (16.4%)	You will enjoy it if you like %GENRE%
Movie Features	Plot (31%)	Here is what the movie is about: %PLOT%.
Movie Features	Year (9.1%)	It premiered in %YEAR%
Movie Features	Avg Rating (16.4%)	It has average score %AVG SCORE%....
Third Party	Broad (7.2%)	Many people seem to like it.
Third Party	Specific (5.5%)	My friend told me it was very good.

3.1 Baselines

We used two out of four template categories described in [16], as well as testing no justification at all. Movie-Feature templates use attributes such as director, cast, genre, *etc.* comprising twenty-four templates in six subcategories. We also use generic Third-Party Opinions, *e.g.*, "A lot of people liked it", with eight templates in two subcategories. Table 1 shows the categories, subcategories, and their distribution from [16].

We tested both *non-contextualized* and *contextualized* template-based justifications. For non-contextual justifications, we select 2 templates according to distribution in Table 1 and populate them using recommended item's metadata, and join the two to return as the justification. For contextual, template-based justifications, we rank the templates described above according to the conversation to create a contextualized justification. We first extract keywords from the conversation and calculate the similarity between those and a list of manually defined aspects corresponding to the templates. Our similarity function is a smoothed inverse of Euclidean distance between Universal Sentence Encoder (USE) [3] embeddings of those keywords and aspects. For example, words like "story", "plot" or "interesting" are most relevant to the plot-related template, while words like "actress", "cast" or "starring", as well as names of all the actors in the recommended movie, are most relevant to a cast-related template. We finally select the top two distinct templates, concatenate them, and return the result as the final justification.

3.2 Overview of out proposed method CONJURE

A natural way to create justifications is by extracting user opinions from a rich set of written reviews [10, 26, 27]. To do that, we extract portions of reviews that contain an opinion about the product or its features. In [15], the authors used existing tools [22] to break the text into EDUs [11]. As an example, consider this review (segments are denoted with braces, and **bold** text marks EDUs suitable for use in justifications):

> "{**this is a timeless movie**}, {it really does age}. {**kubrick was the perfect director**}{to capture stephen king's vision}"

The CONJURE method consists of four basic steps:

(1) Prepare EDUs:
- Extract and encode Elementary Discourse Units (EDUs) [11] from historical reviews associated with the recommended item
- Classify whether each EDU is suitable to be used in justification generation

(2) Process the conversation into the user representation

[1] Note we do not consider the recommender's results. We are only concerned here with the quality of the justification.

(3) Rank the positively classified EDUs from step 1 against the user representation from step 2
(4) Construct candidate justifications and select the most natural (with the lowest perplexity) using the pre-trained OpenAI GPT2 language model [18]
  - If there are at least 2 different EDUs with positive scores, we use them to construct candidates
  - If there is exactly one EDU available, we pick it, and one top template (called "fallback template") ranked using the contextualized baseline approach (§3.1), and use those two to construct candidates
  - If there are no EDUs available (if there are too few reviews and no viable EDUs exist, or if all EDUs have similarity 0), we "fallback" again and return a justification constructed using contextualized baseline approach (§3.1)

For step one: We used existing tools to extract EDUs [22]. For classifying EDUs we iteratively re-trained the model from [15] by adding mislabeled samples to the training data until satisfactory results were achieved. In addition to 1.6k manually labeled samples from the original paper, we have added 330 samples from movie-domain reviews. Resulting classifier achieved 0.95 accuracy and 0.91 F_1 using 5-fold cross-validation.

In this paper we are exploring ways to perform steps two and three, i.e. how to represent the user given the conversation, and how to rank EDUs against that representation. We detail our experiments in §4.2. The key difference between past works and ours is that CONJURE attempts to build justifications without the current user having written any relevant reviews from which we could learn their preferences. CONJURE tries to generate a contextualized justification by ranking EDUs against the current conversation only.

4 EXPERIMENTS

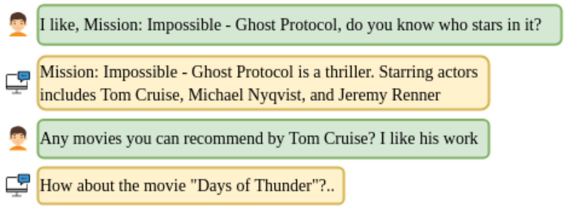
In this section we describe the movie-domain conversations we used to test justification generation approaches, the different conversation representations and EDU ranking options we tested, the metrics we chose to evaluate the resulting justifications, and the design and analysis of a crowd-sourcing task to measure those metrics.

4.1 Conversation corpus

We used conversations from OpenDialKG [13], filtered to include only those about movies and only those with at least one recommended item. We manually extracted movie names from the conversations and crawled `amazon.com` for matching movie titles in their Prime Video department. We identified unique IDs for 450 conversations and downloaded the reviews for those unique IDs.

4.2 Experimental settings and hyperparameters

Process the conversation into the user representation (Step 2). We hypothesized that how we represent the user preferences, as implied in a conversation, would be crucial to constructing an effective justification for the recommended item. We tested three popular methods to extract keywords or phrases from the conversation, in an attempt to focus the EDU ranking on the most salient information: using the Python Keyphrase Extraction package (PKE) [2]; using phrase-level sentiment analysis (Sentires) [27, 28]; or using the whole utterance (no extraction). Additionally, we tested whether to include only user utterances or both user and agent utterances; this had no measurable effect so we do not discuss it further.



I like, Mission: Impossible - Ghost Protocol, do you know who stars in it?

Mission: Impossible - Ghost Protocol is a thriller. Starring actors includes Tom Cruise, Michael Nyqvist, and Jeremy Renner

Any movies you can recommend by Tom Cruise? I like his work

How about the movie "Days of Thunder"?..

System	Justification
static_template	It was released in 1990. It has average score 4.7 out of 5 on Amazon Prime.
conv_template	Tom Cruise stars in this movie. I think you will enjoy it if you like Drama movies.
universal_concat_ _user_keywords	Here is what reviewers said about it: "a young Tom Cruise flashes that billion dollar smile". They also said: "what's not to love. the best Cruise movie ever"
bm25_sep_user_noasp	Here is what reviewers said about it: "great movie. a classic Tom Cruise". They also said: "the film could have done more with Cruise character"

Figure 1: Examples of justifications generated by different methods for a conversation to evaluate justifications

Ranking EDUs against conversation representation (Step 3). Given a set of utterances/keywords/aspects (we call them elements) that represent the conversation, we test whether it is better to find EDUs that best match each element individually and aggregate in the end, or is it better to find EDUs that best match all elements at once.

If we treat them individually (*Separate*), we rank EDUs for each element separately, choose the top EDU for each, and return the EDUs with the highest overall similarity score. This approach has the benefit of addressing different points and possibly returning more diverse EDUs.

If we treat them as one (*Concat*), we concatenate all elements into a single text and compute the similarity of EDUs to that text. This approach benefits when not all aspects have corresponding EDUs, and can potentially return more results.

We have also tested two metrics to measure similarity between EDUs and elements. First, we use BM25 [19] using the Bag-of-Words approach. Second, we compute a Euclidean distance between USE vectors for the conversation, and the templates or EDUs. BM25 has the advantage of returning 0 similarity between texts that have no words in common, while USE always returns a non-zero score. And USE has the advantage of returning an embedding vectors, so it can correctly compare synonims, and does not rely on a string match and common words.

Figure 1 shows one conversation from our data with example justifications generated using different methods.

4.3 Metrics

A comprehensive set of metrics for justifications has been described previously [21], and [1] further explored how to measure whether a given explanation or justification meets a particular goal. Out of seven metrics described in those works, we chose two which most fit our goals: *Efficiency* and *Transparency*. Efficiency, as defined in [21], measures whether a justification helps a user make a decision more quickly. Transparency measures how well the user understands how the system works, that is, how the recommendation was generated. Other metrics either do not fit the conversational movie recommendations setting, or correlate with one of these two.

4.4 Annotation design and analysis

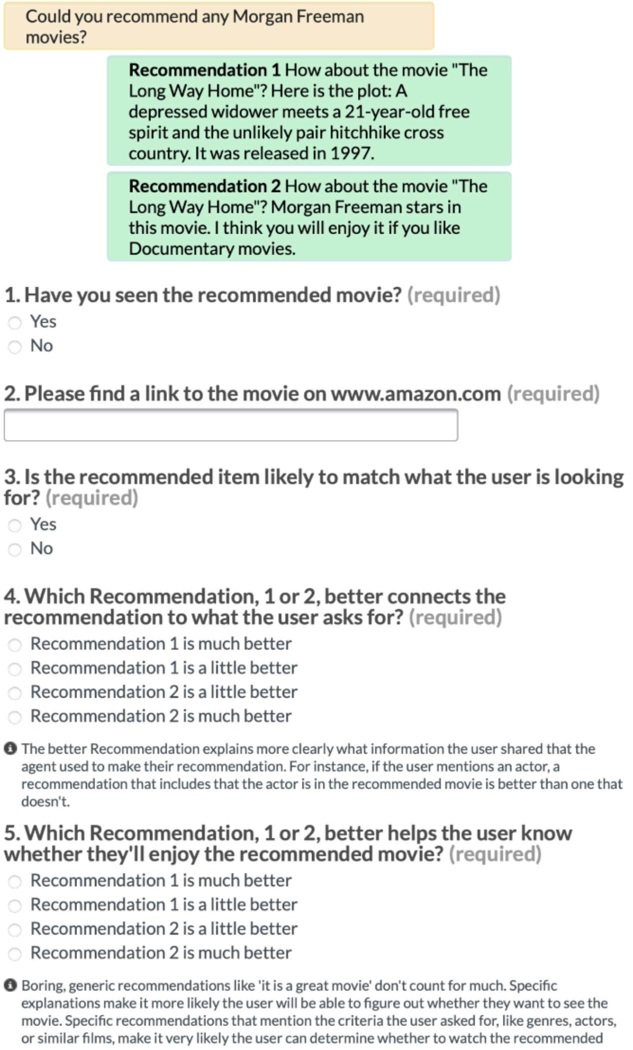
We evaluate all justification configurations discussed in §3 using a pairwise crowd-sourced task, shown in Figure 2. Previous attempts to distinguish between the different CONJURE options using point-wise ratings hinted at trends but failed to reveal anything statistically significant. Pairwise experiments force annotators to make a choice and therefore can reveal preferences in the presence of significant overall variation. Our task includes screening questions to ensure our annotators are engaged in the task and are not bots. Since it would be prohibitively costly to annotate all pairs of conversations, to minimize both cost and estimated model parameter variance we followed [7] to design the experiment and choose pairs for annotators to compare. The design optimizes the chosen pairs of conditions and their order. After filtering the results we ended up with 347 unique pairs of setups covering 194 conversations and 528 justifications annotated by 67 judges. The average directional (i.e., preferring one over the other, regardless of preference strength) inter-rater reliability calculated using Cohen's Kappa was 0.71 for Efficiency and 0.64 for Transparency.

Following [7] we analyze the data using a linear paired comparison model: $Y = \delta + (\mathbf{x}_{1i} - \mathbf{x}_{2i})'\beta$, where $Y \in [-2, 2]$ is the observed score for the pair, δ accounts for positional bias due only to the task layout[2], $\mathbf{x}_{ki}$ are feature vectors corresponding to justification $k \in \{1, 2\}$, and β is a parameter vector estimated by ordinary least squares regression. In this model, a positive coefficient indicates annotators prefer justifications with the corresponding feature. In addition to the four configuration options, we compute two other features. If the system is EDU-based, we recorded how the justification was constructed according to Step 4 (§3.2): whether the system fell back to templates, and if so, whether one EDU was replaced or both (possible values: "not edu", "not fallback", "one template", "two templates"). Here "fallback" means using the template answer when it is not possiblet for some reason to use EDU-based justification (not enough EDUs/similarity is 0/etc). We also include a binary feature indicating whether conversation and justification share named entities.

Alternately, when comparing the baselines which do not have the features of the CONJURE systems, we computed a one-hot feature for each CONJURE system. Finally, we bootstrap the data and report the averaged from 1000 runs results and directly computed p-values.

5 RESULTS AND DISCUSSION

This this section we show the results of our task, compare our model to the baselines, and provide a feature-wise analysis of the proposed method.



Could you recommend any Morgan Freeman movies?

Recommendation 1 How about the movie "The Long Way Home"? Here is the plot: A depressed widower meets a 21-year-old free spirit and the unlikely pair hitchhike cross country. It was released in 1997.

Recommendation 2 How about the movie "The Long Way Home"? Morgan Freeman stars in this movie. I think you will enjoy it if you like Documentary movies.

1. Have you seen the recommended movie? (required)

○ Yes
○ No

2. Please find a link to the movie on www.amazon.com (required)

3. Is the recommended item likely to match what the user is looking for? (required)

○ Yes
○ No

4. Which Recommendation, 1 or 2, better connects the recommendation to what the user asks for? (required)

○ Recommendation 1 is much better
○ Recommendation 1 is a little better
○ Recommendation 2 is a little better
○ Recommendation 2 is much better

ℹ The better Recommendation explains more clearly what information the user shared that the agent used to make their recommendation. For instance, if the user mentions an actor, a recommendation that includes that the actor is in the recommended movie is better than one that doesn't.

5. Which Recommendation, 1 or 2, better helps the user know whether they'll enjoy the recommended movie? (required)

○ Recommendation 1 is much better
○ Recommendation 1 is a little better
○ Recommendation 2 is a little better
○ Recommendation 2 is much better

ℹ Boring, generic recommendations like 'it is a great movie' don't count for much. Specific explanations make it more likely the user will be able to figure out whether they want to see the movie. Specific recommendations that mention the criteria the user asked for, like genres, actors, or similar films, make it very likely the user can determine whether to watch the recommended

Figure 2: The crowd task used to evaluate our justifications.

5.1 Comparing CONJURE systems to baselines

We first compared CONJURE systems to our baselines modeling each CONJURE variant or baseline with its own one-hot feature. Table 2 shows the coefficients associated with each baseline or variation of the CONJURE method. Coefficients are referenced to the baseline which uses *no justification*. Negative values indicate justifications that are *preferred less* than no justification at all.

Static templates (static_template) universally performed worst for both Efficiency and Transparency. Unexpectedly, Conversational Templates (conv_template) were second worst, despite our earlier point-wise tests (not shown) suggesting better performance than some EDU-based systems. In all, eight EDU-based systems significantly outperformed baselines for Efficiency and three for Transparency. In both cases, the best systems use the USE encoding for EDU ranking.

[2] Even though the design [7] includes choosing the order of pairs, explicitly modeling this bias improves the statistical efficiency

Table 2: Statistically Significant regression coefficients for systems-based analysis (higher is better)

Metric	Concat	User	Aspects	Avg Coef	StdErr	p-value
Efficiency						
Intercept (δ)				-0.332	0.078	<0.01
static_template				-0.372	0.121	<0.01
conv_template				-0.321	0.147	<0.01
bm25	concat	user	no aspects	-0.213	0.077	0.01
bm25	separate	user	no aspects	-0.174	0.085	0.04
bm25	concat	both	PKE	-0.131	0.079	0.04
bm25	separate	both	keywords	0.150	0.089	0.04
USE	separate	user	no aspects	0.218	0.115	0.04
USE	separate	user	keywords	0.284	0.169	0.03
USE	concat	user	keywords	0.304	0.144	0.03
USE	**concat**	**both**	**keywords**	**0.342**	**0.144**	**0.01**
Transparency						
Intercept (δ)				-0.401	0.091	<0.01
static_template				-0.389	0.121	<0.01
bm25	separate	user	keywords	0.151	0.079	0.02
USE	**concat**	**user**	**keywords**	**0.254**	**0.148**	**0.04**

5.2 Feature-based analysis

We further explored which features are best for EDU-based systems. The results for feature-based experiments are shown in Table 3. We tried several other parameterizations including interaction terms, and all give similar results. As noted above, we explicitly model the (substantial) bias towards the first item presented, which allows us to achieve the same statistical power with many fewer annotations.

Efficiency. Using the *Universal Sentence Encoder* ("USE") metric for ranking EDUs has a large positive effect (0.29 out of 2). We had hypothesized that if the user has requested a movie with a particular actor, director, etc., that including the entity alone would be enough for the user to make a decision. Extracting *keywords* to represent the conversation ("keywords") was the best approach, a bit better than simply using *all text* ("no aspects"), compared to the *reference condition* (extracting key-phrases with "PKE"). We found no statistically significant interaction between the ranking metric and aspect extraction, indicating that keyword extraction helps even when we embed with USE, which is surprising. Overall, using USE, keyword extraction from the conversation, and using one *fallback template* ("fallback templates: 1") would result in an average preference of ∼ 0.8 over the alternatives.

Transparency. The use of *fallback templates* ("fallback templates: 1") also significantly improves Transparency (0.305 out of 2), which we suspect is in part because these templates tend to use named entities from the conversation. If the recommendation system mentions entities mentioned by the user, it is a clear signal the system has understood the user and is making the recommendation based on their stated preferences. To test this in the future, it would be interesting to experiment with filling in the templates with entities from the films other than those the user mentioned. Again, the *Universal Sentence Encoder* ("USE") was best at ranking EDUs to best match the conversation.

6 CONCLUSIONS

In this work we have proposed a new method for generating Conversational Justifications for Recommendations (CONJURE). We have explored the ways to represent users through conversations instead of more traditionally used historic reviews or numeric scores. We addressed the challenge of introducing factually incorrect information by constructing justifications using a combination of contextually selected templates and Elementary Discourse Units (EDUs) [11] extracted from external movie reviews.

Table 3: Statistically significant regression coefficients for feature-based analysis (higher is better)

Feature	Avg Coef	StdErr	p-value
Efficiency			
Intercept (δ)	-0.314	0.089	<0.001
USE	0.297	0.088	<0.001
keywords	0.304	0.093	<0.001
no aspects	0.207	0.096	0.016
fallback templates: 1	0.176	0.105	0.045
Transparency			
Intercept (δ)	-0.377	0.090	<0.001
USE	0.246	0.093	0.004
fallback templates: 1	0.305	0.103	<0.001

We have shown that even in the absence of any user profile, we can make justifications in a conversational setting, which significantly improve upon template-based or generic baselines, by finding review fragments (EDUs) which better address stated user interests. How the fragments are ranked against the conversation is important; in our case, the best ranking was given by the Universal Sentence Encoder similarity metric and representing the conversation using extracted keywords. More surprising, we found using a combination of review fragments and entity-based templates was beneficial: while purely template-based approaches did not perform well, we suspect templates' reliance on named entities grounds the justification in the conversation, especially when there may be insufficient high-quality review text to rely on. We found no preference for ranking EDUs against all aspects/keywords combined or individually.

We will continue to explore other factors and techniques that might improve our justifications in this important case where we lack extensive user history. One might have guessed that the best-performing keyword-extraction approach would have worked better with BM25 ranking rather than USE, and this may merit further investigation. Further work may also clear up whether users prefer a mix of review- and template-based justifications, or if we simply did not have enough high-quality review content to support all recommendations here. We are also interested to learn whether annotators who have or have not seen the recommended movie might give us different insights into how the justifications are perceived by users, for which we do not yet have sufficient data.

ACKNOWLEDGMENTS

This work was partially supported by a grant from Amazon Alexa towards the study of conversational recommendations and justifications.

REFERENCES

[1] Krisztian Balog and Filip Radlinski. 2020. Measuring Recommendation Explanation Quality: The Conflicting Goals of Explanations. In *Proceedings of the 43rd*

International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20).

[2] Florian Boudin. 2016. PKE: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. Osaka, Japan, 69–73. http://aclweb.org/anthology/C16-2015

[3] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* (2018).

[4] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. 2016. Learning to Rank Features for Recommendation over Multiple Categories. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Pisa, Italy) *(SIGIR '16)*. Association for Computing Machinery, New York, NY, USA, 305–314. https://doi.org/10.1145/2911451.2911549

[5] Felipe Costa, Sixun Ouyang, Peter Dolog, and Aonghus Lawlor. 2018. Automatic Generation of Natural Language Explanations. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion* (Tokyo, Japan) *(IUI '18 Companion)*. Association for Computing Machinery, New York, NY, USA, Article 57, 2 pages. https://doi.org/10.1145/3180308.3180366

[6] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. 2013. How should I explain? A comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies* 72 (01 2013). https://doi.org/10.1016/j.ijhcs.2013.12.007

[7] Peter Goos and Heiko Großmann. 2011. Optimal design of factorial paired comparison experiments in the presence of within-pair order effects. *Food Quality and Preference* 22, 2 (2011), 198–204. https://doi.org/10.1016/j.foodqual.2010.09.008

[8] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1661–1670.

[9] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining Collaborative Filtering Recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (Philadelphia, Pennsylvania, USA) *(CSCW '00)*. Association for Computing Machinery, New York, NY, USA, 241–250. https://doi.org/10.1145/358916.358995

[10] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 345–354.

[11] William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text* 8, 3 (1988), 243–281.

[12] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th ACM Conference on Recommender Systems* (Hong Kong, China) *(RecSys '13)*. Association for Computing Machinery, New York, NY, USA, 165–172. https://doi.org/10.1145/2507157.2507163

[13] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 845–854.

[14] Cataldo Musto, Gaetano Rossiello, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2019. Combining Text Summarization and Aspect-Based Sentiment Analysis of Users' Reviews to Justify Recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 383–387. https://doi.org/10.1145/3298689.3347024

[15] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 188–197.

[16] Florian Pecune, Shruti Murali, Vivian Tsai, Yoichi Matsuyama, and Justine Cassell. 2019. A Model of Social Explanations for a Conversational Movie Recommendation System. In *Proceedings of the 7th International Conference on Human-Agent Interaction* (Kyoto, Japan) *(HAI '19)*. Association for Computing Machinery, New York, NY, USA, 135–143. https://doi.org/10.1145/3349537.3351899

[17] Gustavo Penha and Claudia Hauff. 2020. What does BERT know about books, movies and music? Probing BERT for Conversational Recommendation. *CoRR* abs/2007.15356 (2020). arXiv:2007.15356 https://arxiv.org/abs/2007.15356

[18] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[19] Stephen E Robertson. 1977. The probability ranking principle in IR. *Journal of documentation* (1977).

[20] Rashmi Sinha and Kirsten Swearingen. 2002. The Role of Transparency in Recommender Systems. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems* (Minneapolis, Minnesota, USA) *(CHI EA '02)*. Association for Computing Machinery, New York, NY, USA, 830–831. https://doi.org/10.1145/506443.506619

[21] Nava Tintarev and Judith Masthoff. 2015. Explaining recommendations: Design and evaluation. In *Recommender systems handbook*. Springer, 353–382.

[22] Yizhong Wang, Sujian Li, and Jingfeng Yang. 2018. Toward fast and accurate neural discourse segmentation. *arXiv preprint arXiv:1808.09147* (2018).

[23] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep language-based critiquing for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 137–145.

[24] Yao Wu and Martin Ester. 2015. FLAME: A Probabilistic Model Combining Aspect Based Opinion Mining and Collaborative Filtering. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining* (Shanghai, China) *(WSDM '15)*. Association for Computing Machinery, New York, NY, USA, 199–208. https://doi.org/10.1145/2684822.2685291

[25] L. Richard Ye and Paul E. Johnson. 1995. The Impact of Explanation Facilities on User Acceptance of Expert Systems Advice. *MIS Quarterly* 19, 2 (1995), 157–172. http://www.jstor.org/stable/249686

[26] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101. https://doi.org/10.1561/1500000066

[27] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 83–92.

[28] Yongfeng Zhang, Haochen Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. 2014. Do Users Rate or Review? Boost Phrase-Level Sentiment Labeling with Review-Level Sentiment Classification. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval* (Gold Coast, Queensland, Australia) *(SIGIR '14)*. Association for Computing Machinery, New York, NY, USA, 1027–1030. https://doi.org/10.1145/2600428.2609501