

# Textual Explanations and Critiques in Recommendation Systems

THIS IS A TEMPORARY TITLE PAGE  
It will be replaced for the final print by a version  
provided by the registrar's office.

Thèse n. 9047

Présentée le 25 février 2022

à la Faculté informatique et communications

Laboratoire d'intelligence artificielle

Programme doctoral en informatique et communications

École polytechnique fédérale de Lausanne

pour l'obtention du grade de Docteur ès Sciences

par

Diego Matteo ANTOGNINI

Acceptée sur proposition du jury:

Prof. R. West, président du jury

Prof. B. Faltings, directeur de thèse

Prof. J. McAuley, rapporteur

Prof. S. Sanner, rapporteur

Prof. A. Bosselut, rapporteur

Lausanne, EPFL, 2022





Our greatest weakness lies in giving up. The most certain  
way to succeed is always to try just one more time.  
— Thomas A. Edison

To my family, for their love and support...



# Acknowledgements

First of all, I would like to express my deepest gratitude to my thesis advisor, Prof. Boi Faltings, for his insightful supervision, support, flexibility, and scientific freedom to pursue my research interests. I am also grateful to Dr. Claudiu Musat for always being so positive and encouraging me to always give my best and never give up. I would like to thank Prof. Julian McAuley for his guidance and care during my research stay at the University of California, San Diego.

I had the honor of having Prof. Robert West (EPFL), Prof. Julian McAuley (UCSD), Prof. Scott Sanner (University of Toronto), and Prof. Antoine Bosselut (EPFL) for my thesis committee. Thank you for taking the time to read my thesis, participate in my thesis defense and for your insightful feedback and helpful suggestions on the thesis.

During my Ph.D. journey, I spent a great time with current and former members of the artificial intelligence laboratory (LIA): Panayiotis, Aleksei, Igor, Ljubomir, Fei, and also Naman, Zeki, and Adam. Special thanks to Aleksei for all the great times in San Diego. When we were not surfing, we were climbing one of the highest peaks in the U.S.A. in 13 hours (Mount Whitney, 4,421m), eating great food, or simply going through the Death Valley or cathedral rocks in Sedona. Thank you also to Panayiotis for the great times and discussions. Thank you to Damien Doy for making me discover hike challenges: around 40km in 14 hours from Champéry (1,050m) - Dents du Midi (3,258) - Evionnaz (469m). A special thanks to Alexandre Perez for bike tours, restaurants, and other great times. Also, I would like to thank our former secretaries Sylvie Thomet and Patricia Defferrard for their tireless help with all the administrative services. Also, special thanks to Nichole Loyd Janlov who helped me so much with all the administrative steps to come to UCSD. Finally, the IT support of EPFL made sure IT problems were minimal and promptly resolved. Thank you to Stéphane Ecuyer, Carlos Perez, and Yohan Moulin.

A heartfelt thank you to the 30+ students that I had the privilege of supervising during my Ph.D.: Kirtan Padh, Saibo Geng, Shuangqi Li, Antoine Scardigli, Milena Filipovic, Blagojce Mitrevski, Martin Milenkoski, Nikola Milojkovic, and many more. Although a huge thank you to Prof. Hatem Ghorbel and Prof. Cédric Billat for great discussions and for asking me to assess many students' bachelor and master theses. Same for Pierre Ferrari and Renato Rota for CFC/EFZ final projects.

I am very thankful to my parents, Luciano and Ana, and my siblings, Sergio and Laura, and my nephew Noa, for their support and encouragement. Finally, I wish to thank my wonderful fiancée, Michaela, for her support and patience, but most importantly for her love, her person, all the delicious meals and desserts she made, and all the great adventures we had together!

*Lugano, January 26, 2023*

D. A.



# Abstract

Artificial intelligence and machine learning algorithms have become ubiquitous. Although they offer a wide range of benefits, their adoption in decision-critical fields is limited by their lack of interpretability, particularly with textual data. Moreover, with more data available than ever before, it has become increasingly important to explain automated predictions.

Generally, users find it difficult to understand the underlying computational processes and interact with the models, especially when the models fail to generate the outcomes or explanations, or both, correctly. This problem highlights the growing need for users to better understand the models' inner workings and gain control over their actions. This dissertation focuses on two fundamental challenges of addressing this need. The first involves *explanation generation*: inferring high-quality explanations from text documents in a scalable and data-driven manner. The second challenge consists in making explanations actionable, and we refer to it as *critiquing*. This dissertation examines two important applications in natural language processing and recommendation tasks.

First, we present two models for extracting high-quality and personalized justifications from text documents under the paradigm of selective rationalization – a binary selection of input features from which the model computes the outcome. We propose the concept of multi-dimensional rationales and emphasize that a single overall selection does not accurately capture the multi-faceted nature of useful rationales. The first method relies on multi-task learning and learns the rationales in an unsupervised manner, with no prior in the data. Inspired by the role of concept-based thinking in human reasoning, our second model is a generalization: it assumes that only one label is observed. We empirically demonstrate the efficiency of the models in terms of explanation and predictive performance as well as their scalability. It opens new doors for applications, such as recommendation or summarization. Second, we use the high-level idea of multi-dimensional rationalization to mine massive corpora and construct large personalized justification datasets. We show that human users significantly prefer our explanations over those produced by state-of-the-art methods.

Last, we propose two conversational explainable recommendation systems. The first explains a user rating by inferring a set of keyphrases. Conditioned on them, it generates an abstractive personal justification. We allow users to interact iteratively with the explanations and refine the recommendation through an unsupervised critiquing method. Our second model is based on multimodal modeling and self-supervision. It enables fast and efficient multi-step critiquing. Using real-world datasets, we show that our models exhibit better performance than state-of-the-art models in terms of recommendation, explanation, and critiquing.

## Abstract

---

Overall, we demonstrate that interpretability does not come at the cost of reduced performance in two consequential applications. Our framework is applicable to other fields as well. This dissertation presents an effective means of closing the gap between promise and practice in artificial intelligence.

**Keywords:** explainability, interpretability, deep learning, selective rationalization, conversational explainable recommendation systems, multi-step critiquing, variational autoencoder



# Résumé

L'intelligence artificielle et les algorithmes d'apprentissage automatique sont devenus omniprésents. Bien qu'ils offrent un large éventail d'avantages, leur adoption dans les domaines critiques pour la décision est limitée par leur manque d'interprétabilité, en particulier avec les données textuelles. De plus, comme les données disponibles sont plus nombreuses que jamais, il est devenu de plus en plus important d'expliquer les prédictions automatisées.

En général, les utilisateurs ont du mal à comprendre les processus de calcul sous-jacents et à interagir avec les modèles, surtout lorsque ces derniers ne parviennent pas à générer correctement les résultats ou les explications, ou les deux. Ce problème met en évidence le besoin croissant des utilisateurs de mieux comprendre le fonctionnement interne des modèles et d'avoir un contrôle sur leurs actions. Cette thèse se concentre sur deux défis fondamentaux pour répondre à ce besoin. Le premier concerne la *génération d'explications* : inférer des explications de haute qualité à partir de documents textuels à grande échelle et d'une manière orientée données. Le second défi consiste à rendre les explications interactives, ce que nous appelons *critiquer*. Cette thèse examine deux applications importantes dans le traitement du langage naturel et la recommandation.

Tout d'abord, nous présentons deux modèles pour extraire des justifications personnalisées et de haute qualité à partir de documents textuels sous le paradigme de la rationalisation sélective - une sélection binaire de caractéristiques d'entrée à partir desquelles le modèle calcule le résultat. Nous proposons le concept de justifications multidimensionnelles et montrons qu'une seule sélection globale ne permet pas d'identifier avec précision la nature multidimensionnelle des justifications utiles. La première méthode repose sur l'apprentissage multi-tâches et apprend à extraire les justifications de manière non supervisée, sans assumptions sur les données. Inspiré par le rôle de la pensée basée sur les concepts dans le raisonnement humain, notre deuxième modèle en est une généralisation : il suppose qu'une seule étiquette est observée. Nous démontrons empiriquement l'efficacité des modèles en termes de performances d'explication et de prédiction, ainsi que leur évolutivité. Cela ouvre de nouvelles portes pour des applications telles que la recommandation ou le résumé.

Deuxièmement, nous utilisons l'idée de haut niveau de la rationalisation multidimensionnelle pour exploiter des corpus massifs et construire de grands ensembles de données de justifications personnalisées. Nous montrons que les utilisateurs humains préfèrent nettement nos explications à celles produites par les méthodes les plus récentes.

Enfin, nous proposons deux systèmes de recommandation conversationnels et explicables. Le premier explique une évaluation d'utilisateur en déduisant un ensemble de mots-clés. En

fonction de ceux-ci, il génère une justification personnalisé. Nous permettons aux utilisateurs d'interagir itérativement avec les explications et d'affiner la recommandation par une méthode de critique non supervisée. Notre deuxième modèle est basé sur la modélisation multimodale et l'auto-supervision. Il permet une critique rapide et efficace en plusieurs étapes. En utilisant des ensembles de données réelles, nous montrons que nos modèles présentent de meilleures performances que les modèles de pointe en termes de recommandation, d'explication et de critique.

Dans l'ensemble, nous démontrons que l'interprétabilité ne se fait pas au prix d'une réduction des performances dans deux applications conséquentes. Notre cadre est également applicable à d'autres domaines. Cette thèse présente un moyen efficace de combler le fossé entre les promesses et la pratique en intelligence artificielle.

**Mots-clés** : explicabilité, interprétabilité, apprentissage profond, rationalisation sélective, systèmes de recommandation explicables et conversationnels, critiquer, auto-encodeur variationnel

# Publications

The contributions and the results presented in this thesis were published in the following publications:

1. **Antognini et al. (2021b):** Antognini, D., Musat, C., and Faltings, B. (2021b). Multi-dimensional explanation of target variables from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence: AAAI 2021*, 35(14):12507–12515.
2. **Antognini and Faltings (2021b):** Antognini, D. and Faltings, B. (2021b). Rationalization through concepts. In *Findings of the Association for Computational Linguistics: ACL 2021*, pages 761–775, Online. Association for Computational Linguistics.
3. **Antognini et al. (2021a):** Antognini, D., Musat, C., and Faltings, B. (2021a). Interacting with explanations through critiquing. In Zhou, Z.-H., editor, In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence: IJCAI 2021*, pages 515–521. International Joint Conferences on Artificial Intelligence Organization. Main Track.
4. **Antognini and Faltings (2021a):** Antognini, D. and Faltings, B. (2021a). Fast multi-step critiquing for vae-based recommender systems. In *Fifteenth ACM Conference on Recommender Systems: RecSys 2021*, page 209–219, New York, NY, USA. Association for Computing Machinery.
5. **Petrescu et al. (2021):** \*Petrescu, D. A., \*Antognini, D., and Faltings, B. (2021). Multi-step critiquing user interface for recommender systems. In *Fifteenth ACM Conference on Recommender Systems: RecSys 2021*, page 760–763, New York, NY, USA. Association for Computing Machinery.

The following publications were part of the research conducted during the Ph.D.. They present results that are supplementary to or build upon but are not covered in this dissertation:

6. **Antognini et al. (2023):** Antognini, D., Li, S., Faltings, B., and McAuley, J. (2023). Assistive recipe editing through critiquing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online. Association for Computational Linguistics.
7. **Antognini and Faltings (2022):** Antognini, D. and Faltings, B. (2022). Positive & Negative Critiquing for VAE-based Recommenders. arXiv preprint arXiv:2204.02162.

8. **Padh et al. (2021):** Padh, K., Antognini, D., Lejal Glaude, E., Faltings, B., and Musat, C. (2021). Addressing fairness in classification with a model-agnostic multi-objective algorithm. In Peters, J. and Sontag, D., editors, In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence (UAI): UAI 2021*, Proceedings of Machine Learning Research. PMLR.
9. **Antognini and Faltings (2020b):** Antognini, D. and Faltings, B. (2020b). HotelRec: a novel very large-scale hotel recommen- dation dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference: LREC 2021*, pages 4917–4923, Marseille, France. European Language Resources Association.
10. **Antognini and Faltings (2020a):** Antognini, D. and Faltings, B. (2020a). GameWikiSum: a novel large multi-document summarization dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference: LREC 2020*, pages 6645–6650, Marseille, France. European Language Resources Association.
11. **Antognini and Faltings (2019):** Antognini, D. and Faltings, B. (2019). Learning to create sentence semantic relation graphs for multi-document summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 32–41, Hong Kong, China. Association for Computational Linguistics.
12. **Milojkovic et al. (2020):** Milojkovic, N., Antognini, D., Bergamin, G., Faltings, B., and Musat, C. (2020). Multi-gradient descent for multi-objective recommender systems. In Proceedings of the AAAI (2020) - Workshop on Interactive and Conversational Recommendation Systems (WICRS).
13. **Filipovic et al. (2021):** Filipovic, M., Mitrevski, B., Antognini, D., Lejal Glaude, E., Faltings, B., and Musat, C. (2021). Modeling online behavior in recommender systems: The importance of temporal context. In Proceedings of the Perspectives on the Evaluation of Recommender Systems Workshop at RecSys 2021 (PERSPECTIVE 2021).
14. **Milenkoski et al. (2021):** Milenkoski, M., Antognini, D., and Musat, C. (2021). Recommending burgers based on pizza preferences: Addressing data sparsity with a product of experts. In Proceedings of the First Workshop of Cross-Market Recommendation at RecSys 2021 (XMRec 2021).
15. **Mitrevski et al. (2021):** Mitrevski, B., Filipovic, M., Antognini, D., Lejal Glaude, E., Faltings, B., and Musat, C. (2021). Momentum-based gradient methods in multi-objective recommendation. In Proceedings of the First Workshop on Multi-Objective Recommender Systems at RecSys 2021 (MORS 2021).
16. **Giannakopoulos et al. (2017):** \*Giannakopoulos, A., \*Antognini, D., Musat, C., Hossmann, A., and Baeriswyl, M. (2017). Dataset construction via attention for aspect term extraction with distant supervision. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW): ICDMW 2017*, pages 373–380. IEEE.

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract (English/Français)</b>	<b>iii</b>
<b>Publications</b>	<b>vii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Dissertation Structure and Contributions . . . . .	5
<b>I Extracting Multi-Faceted Explanations from Text Documents</b>	<b>9</b>
<b>2 Multi-Dimensional Rationalization</b>	<b>11</b>
2.1 Preface . . . . .	11
2.2 Introduction . . . . .	12
2.3 Related Work . . . . .	14
2.3.1 Interpretability . . . . .	14
2.3.2 Multi-Aspect Sentiment Classification . . . . .	14
2.3.3 Attention-based Models . . . . .	14
2.3.4 Rationale Models . . . . .	15
2.4 MTM: A Multi-Target Masker . . . . .	16
2.4.1 Model Overview . . . . .	16
2.4.2 Enabling the Interpretability of Masks . . . . .	18
2.5 Experiments . . . . .	18
2.5.1 Experimental Details . . . . .	19
2.5.2 Datasets . . . . .	19
2.5.3 Baselines . . . . .	20
2.6 Results . . . . .	22
2.6.1 Multi-Rationale Interpretability . . . . .	22
2.6.2 Multi-Aspect Sentiment Classification . . . . .	24
2.6.3 Impact of the Length of Rationales . . . . .	27
	ix

2.7	Conclusion . . . . .	29
<b>3</b>	<b>Rationalization through Concepts</b>	<b>31</b>
3.1	Preface . . . . .	31
3.2	Introduction . . . . .	32
3.3	Related Work . . . . .	33
3.3.1	Interpretability . . . . .	34
3.3.2	Explanations through Concepts . . . . .	34
3.3.3	Topic Modeling . . . . .	35
3.4	ConRAT: A Concept-based Rationalizer . . . . .	35
3.4.1	Model Overview . . . . .	36
3.4.2	Unsupervised Discovery of Concepts . . . . .	37
3.4.3	Improving Overall Performance Further . . . . .	38
3.5	Experiments . . . . .	39
3.5.1	Datasets . . . . .	39
3.5.2	Baselines . . . . .	40
3.5.3	Experimental Details . . . . .	40
3.5.4	RQ 1: Can ConRAT find evidence for factual and counterfactual rationales? . . . . .	41
3.5.5	RQ 2: Are concepts inferred by ConRAT consistent with human rationalization? . . . . .	42
3.5.6	RQ 3: How does the number of concepts $K$ in ConRAT affect the performance? . . . . .	45
3.5.7	RQ 4: How does each module of ConRAT contribute to the overall performance? . . . . .	46
3.6	Conclusion . . . . .	46
<b>II</b>	<b>Iterative Critiquing of Explanations</b>	<b>47</b>
<b>4</b>	<b>Generating Explanations and Interacting with Them through Critiquing</b>	<b>49</b>
4.1	Preface . . . . .	49
4.2	Introduction . . . . .	50
4.3	Related Work . . . . .	53
4.3.1	Textual Explainable Recommendation . . . . .	53
4.3.2	Critiquing . . . . .	53
4.4	Extracting Justifications from Reviews . . . . .	54
4.5	T-RECS: A Multi-Task Transformer with Explanations and Critiquing . . . . .	56
4.5.1	Model Overview . . . . .	56
4.5.2	Unsupervised Critiquing . . . . .	58
4.6	Experiments . . . . .	59
4.6.1	Experimental Settings . . . . .	59
4.6.2	RQ 1: Are <i>markers</i> appropriate justifications? . . . . .	60
4.6.3	RQ 2: Does T-RECS infer high-quality and personalized explanations? . . . . .	61

4.6.4	RQ 3: Do T-RECS justifications benefit the overall recommendation quality? . . . . .	65
4.6.5	RQ 4: Can T-RECS enable critiquing? . . . . .	69
4.7	Multi-Step Critiquing User Interface . . . . .	70
4.7.1	System Overview . . . . .	70
4.7.2	Interface Details . . . . .	72
4.7.3	Discussion . . . . .	73
4.8	Conclusion . . . . .	74
<b>5</b>	<b>Faster and Better Multi-Step Critiquing via Self-Supervision</b>	<b>75</b>
5.1	Preface . . . . .	75
5.2	Introduction . . . . .	76
5.3	Preliminaries . . . . .	77
5.3.1	Notation . . . . .	78
5.3.2	Variational Autoencoder for Recommendation (VAE) . . . . .	78
5.3.3	Co-embedding of Language-based Feedback with the Variational Autoencoder (CE-VAE) . . . . .	79
5.4	M&Ms-VAE: A Mixture-of-Experts Multimodal Variational AutoEncoder . . . . .	80
5.4.1	Model Overview . . . . .	80
5.4.2	Training Strategy . . . . .	82
5.4.3	Self-Supervised Critiquing with M&Ms-VAE . . . . .	83
5.5	Experiments . . . . .	86
5.5.1	Datasets . . . . .	86
5.5.2	Experimental Settings . . . . .	87
5.5.3	RQ 1: How does M&Ms-VAE perform in terms of recommendation and explanation performance? . . . . .	87
5.5.4	RQ 2: Can M&Ms-VAE with the self-supervised critiquing objective enable multi-step critiquing? . . . . .	91
5.5.5	RQ 3: What is the critiquing computational time complexity for M&Ms-VAE compared to prior work? . . . . .	94
5.5.6	RQ4: How does M&Ms-VAE perform under weak supervision; is the joint & cross inference coherent? . . . . .	95
5.6	Conclusion . . . . .	97
<b>6</b>	<b>Conclusion</b>	<b>99</b>
6.1	Future Directions . . . . .	101
	<b>Appendices</b>	<b>105</b>
<b>A</b>	<b>Complementary Results</b>	<b>107</b>
A.1	Inferred Concepts by ConRAT . . . . .	107
A.1.1	Objective Evaluation . . . . .	107
A.1.2	Subjective Evaluation . . . . .	107

## Contents

---

A.2	Full Natural Language Explanations Results . . . . .	108
A.3	Full Keyphrase Explanation Results . . . . .	110
A.4	Multi-Step Critiquing - Additional Metrics for T-RECS . . . . .	110
<b>B</b>	<b>Processing</b>	<b>111</b>
B.1	Processing & Filtering <i>Markers</i> . . . . .	111
B.2	Extracting Keyphrase for T-RECS . . . . .	111
B.3	Addressing Users without Reviews in T-RECS . . . . .	112
<b>C</b>	<b>Visualizations</b>	<b>113</b>
C.1	Extracted Topic Words per Aspect from Rationales . . . . .	113
C.2	Multi-Dimensional Rationales . . . . .	116
C.3	Concepts . . . . .	121
<b>D</b>	<b>Human Evaluation Details</b>	<b>123</b>
D.1	Understandability of Concepts . . . . .	123
D.2	Justification Evaluation . . . . .	123
<b>E</b>	<b>M&amp;Ms-VAE Derivation</b>	<b>127</b>
<b>F</b>	<b>Reproducibility</b>	<b>129</b>
F.1	Multi-Target Masker (MTM) . . . . .	129
F.2	Concept-based Rationalizer (ConRAT) . . . . .	130
F.3	A Multi-Task Transformer with Explanations and Critiquing (T-RECS) . . . . .	131
F.4	MultiModal Variational AutoEncoder (M&Ms-VAE) . . . . .	132
	<b>Bibliography</b>	<b>135</b>
	<b>Curriculum Vitae</b>	<b>157</b>



# List of Figures

1.1	The cycle of conversational critiquing. First, the user informs the recommender system of his or her initial preferences. Then, the model returns a list of recommended items with their respective explanations. The user can interact with the explanations and critique them. This process (❷ - ❹) repeats until the user accepts the recommendation and ceases to provide critiques. . . . .	4
2.1	A beer review with explanations produced by an attention model and our Multi-Target Masker model. The colors depict produced rationales (i.e., justifications) of the rated aspects: Appearance, Smell, Taste, and Palate. The induced rationales mostly lead to long sequences that clearly describe each aspect (one switch ★ per aspect), while the attention model has many short, noisy interleaving sequences. . . . .	13
2.2	The proposed Multi-Target Masker ( <i>MTM</i> ) model architecture to predict and explain $T$ target variables. . . . .	16
2.3	Baseline architectures. . . . .	21
2.4	Induced rationales on a truncated hotel review, where shade colors represent the model confidence towards the aspects. <i>MTM</i> finds most of the crucial spans of words with a small amount of noise. <i>SAM</i> lacks coverage but identifies words where half are correct and the others ambiguous (represented with colored underlines). . . . .	26
2.5	Performance related to human evaluation, showing the precision, recall, and F1 scores of the rationales for each aspect (including the overall rating) of the <i>Beer</i> dataset. The percentage of words indicates the number of highlighted words of the full review. . . . .	28
3.1	An illustration of ConRAT. Given a beer review, ConRAT identifies five excerpts that relate to particular concepts of beers (i.e., the explanation), depicted in color, from which it computes the outcome. . . . .	33
3.2	The proposed self-explaining model ConRAT. The model predicts and explains $\hat{y}$ . Given a document $X$ , the concept generator produces one binary mask per concept. The concept selector decides which concepts are present in the input. The predictor aggregates each selected concept's prediction to compute $\hat{y}$ . . . .	35

3.3	Concepts generated (with $\ell=10$ ) for a beer review. <u>Underline</u> highlights ambiguities. The color depicts the aspects: Appearance, Aroma, Palate, Taste, and Overall. <b>ConRAT is trained only on the overall label</b> . RNP produces the same rationale for the aspect Aroma and Palate. InvRAT infers wrong rationales for the aspects Palate and Taste. In contrast, ConRAT reduces the impact of spurious correlations.	43
3.4	Subjective performance of rationales for the multi-aspect beer reviews. Evaluators need to guess both the sentiment and what aspect the concept is about, which makes random guess only 10%. ConRAT infers meaningful rationales according to human annotators.	44
4.1	A flow of conversational critiquing over two time steps. a) The system proposes to the user a recommendation with a keyphrase explanation and a justification. The user can interact with the explanation and critique phrases. b) A new recommendation is produced from the user's profile and the critique. c) This process repeats until the user accepts the recommendation and ceases to provide critiques.	51
4.2	For reviews written by a user $u$ and a set of reviews about an item $i$ , we extract the justifications for each aspect rating and implicitly build an interest profile. T-RECS outputs a personalized recommendation with two explanations: the keyphrases reflecting the overlap between the two profiles, and a synthetic justification conditioned on the latter.	52
4.3	Extracted justifications from a hotel review and a beer review. The inferred <i>markers</i> depict the excerpts that explain the ratings of the aspects: Service, Cleanliness, Value, Room, and Location. We denote in <b>bold</b> the EDU-based justification from the model of (Ni et al., 2019).	54
4.4	(Left) Preprocessing for the users and the items. For each user $u$ and item $i$ , we first extract <i>markers</i> from their past reviews (highlighted in color), using the pretrained multi-target masker (see Section 4.4), that become their respective justifications. Then, we sample $N_{just}$ of them and build the justification references $J^u$ and $J^i$ , respectively. (Right) T-RECS architecture. Given a user $u$ and an item $i$ with their justification references $J^u$ , $J^i$ and latent factors $\beta^u$ , $\beta^i$ , T-RECS produces a joint embedding $\mathbf{z}$ from which it predicts a rating $\hat{\mathbf{y}}_r$ , a keyphrase explanation $\hat{\mathbf{y}}_{kp}$ , and a natural language justification $\hat{\mathbf{y}}_{just}$ conditioned on $\hat{\mathbf{y}}_{kp}$ .	55
4.5	Workflow of considering to recommend items to a user $u$ . We illustrate it for a given item $i$ . <b>Black</b> denotes the forward pass to infer the rating $\hat{\mathbf{y}}_r$ with the explanations $\hat{\mathbf{y}}_{kp}$ and $\hat{\mathbf{y}}_{just}$ . <b>Yellow</b> indicates the critiquing: the user critiques the binary-vector keyphrase explanation $\hat{\mathbf{y}}_{kp}$ (e.g., <i>center</i> ) to $\tilde{\mathbf{y}}_{kp}^*$ , which modulates the latent space into $\mathbf{z}^*$ for each item. <b>Orange</b> shows the new forward pass for the subsequent recommendation $\hat{\mathbf{y}}_r^*$ and explanations $\hat{\mathbf{y}}_{kp}^*$ , $\hat{\mathbf{y}}_{just}^*$ .	58
4.6	Performance of the preference prediction using Kendall's $\tau$ and $\delta =  y_r^i - y_r^j $ .	67
4.7	Single- (top) and multi-step (bottom) critiquing.	69

4.8	Interactions overview. . . . .	71
4.9	Interfaces. . . . .	73
5.1	Probabilistic-graphical-model view of our M&Ms-VAE model. Both the implicit feedback $\mathbf{r}_u$ and the keyphrase $\mathbf{k}_u$ are generated from user $u$ 's latent representation $\mathbf{z}_u$ . Solid lines denote the generative model $p_\Theta$ , whereas dashed lines denote the variational approximation $q_\Phi$ . . . . .	81
5.2	The proposed M&Ms-VAE architecture and training scheme. Each pass infers the parameters $\mu_u$ and $\sigma_u$ with the mixture of experts using either the joint inference network $q_\Phi(\mathbf{z}_u \mathbf{r}_u, \mathbf{k}_u)$ or one of the individual networks ( $q_{\Phi_r}(\mathbf{z}_u \mathbf{r}_u)$ or $q_{\Phi_k}(\mathbf{z}_u \mathbf{k}_u)$ , respectively). The final gradient is computed on the sum of each $ELBO(\cdot)$ term. . . . .	83
5.3	Workflow of considering the recommendation of items to a user $u$ over $t+1$ time steps. First, M&Ms-VAE produces the initial set of recommended items $\hat{\mathbf{r}}_u^0 = \hat{\mathbf{r}}_u$ using only the historical observed interactions $\mathbf{r}_u$ . Then, the user can provide a critique $\mathbf{c}_u^t$ that is encoded into $\mathbf{z}_u^t$ via the inference model $q_{\Phi_k}(\mathbf{z}_u \mathbf{k}_u)$ . The blending module combines the previous representations $\mathbf{z}_u^0, \mathbf{z}_u^1, \dots, \mathbf{z}_u^t$ into $\tilde{\mathbf{z}}_u^t$ , from which the subsequent recommendation $\hat{\mathbf{r}}_u^t$ is computed. This process continues until the user $u$ accepts the recommendation and ceases to provide additional critiques. . . . .	84
5.4	Multi-step critiquing performance after 10 turns of conversation. For each dataset and keyphrase critiquing selection method, we report the average success rate (left y-axis) and session length (right y-axis) at different Top-N with 95% confidence interval. . . . .	93
5.5	Multi-step critiquing performance after 10 turns of conversation <b>on all items</b> . For each dataset and keyphrase critiquing selection method, we report the average success rate (left y-axis) and session length (right y-axis) at different Top-N with 95% confidence interval. . . . .	94
5.6	Average time consumed for completing 10 runs for 1,000-user simulation after ten turns of conversation with 95% confidence intervals. LLC-Score and LLC-Rank cannot leverage GPUs; we thus report the performance of M&Ms-VAE on CPU and GPU. Additionally, we report the inference speedup compared to the slowest model. . . . .	95
5.7	Recommendation (left y-axis) and explanation (right y-axis) results averaged on five runs with different combinations of modalities observed at inference. The x-axis denotes the percentage of fully observed inputs during training; the rest is partially observed. . . . .	96
A.1	Subjective performance per aspect of rationales for the multi-aspect beer reviews.	108
A.2	Multi-step critiquing performance. Keyphrase prediction over multi-step critiquing in terms of Recall@10, NDCG@10, Precision@10, and MAP@10 with 95% confidence interval. a) Results on the hotel dataset, b) on the beer dataset. . . .	110

## List of Figures

---

- C.1 A sample review from the **Beer** dataset, with computed masks from different methods. *MTM* achieves near-perfect annotations, while *SAM* highlights only two words where one is ambiguous with respect to the four aspects. *MAA* mixes between the aspect *Appearance* and *Smell*. *MASA* identifies some words but lacks coverage. . . . . 116
- C.2 A sample review from the **Beer** dataset, with computed masks from different methods. *MTM* can accurately identify what parts of the review describe each aspect. *MAA* provides very noisy labels due to the high imbalance and correlation between aspects, while *MASA* highlights only a few important words. We can see that *SAM* is confused and performs a poor selection. . . . . 117
- C.3 A sample review from the **Hotel** dataset, with computed masks from different methods. *MTM* emphasizes consecutive words, identifies essential spans while having a small amount of noise. *SAM* focuses on certain specific words and spans, but labels are ambiguous. The *MAA* model highlights many words, ignores a few crucial keyphrases, but labels are noisy when the confidence is low. *MASA* provides noisier tags than *MAA*. . . . . 118
- C.4 A sample review from the **Hotel** dataset, with computed masks from different methods. Our *MTM* model finds most of the crucial span of words with a small amount of noise. *SAM* lacks coverage but identifies words where half are correctly tags and the others ambiguous. *MAA* partially correctly highlights words for the aspects *Service*, *Location*, and *Value* while missing out on the aspect *Cleanliness*. *MASA* confidently finds a few important words. . . . . 119
- C.5 A sample review from the **decorrelated Beer** dataset, with computed masks from different methods. Our model *MTM* highlights all the words corresponding to the aspects. *SAM* only highlights the most crucial information, but some words are missing out and one is ambiguous. *MAA* and *MASA* fail to identify most of the words related to the aspect *Appearance*, and only a few words have high confidence, resulting in noisy labeling. Additionally, *MAA* considers words belonging to the aspect *Taste* whereas this dataset does not include it in the aspect set (because it has a high correlation with other rating scores). . . . . 120
- C.6 A sample review from the **decorrelated Beer** dataset, with computed masks from different methods. *MTM* finds the exact parts corresponding to the aspect *Appearance* and *Palate* while covering most of the aspect *Smell*. *SAM* identifies key-information without any ambiguity, but lacks coverage. *MAA* highlights confidently nearly all the words while having some noise for the aspect *Appearance*. *MASA* selects confidently only most predictive words. . . . . 120
- C.7 Examples of generated rationales with  $\ell = 10$  for a beer review. Underline highlights ambiguities. . . . . 121
- C.8 Examples of generated rationales with  $\ell = 20$  for a beer review. Underline highlights ambiguities. . . . . 122

D.1	Annotation platform for judging the quality of the concepts in the subjective evaluation on beer reviews. . . . .	124
D.2	Annotation platform for judging the quality of extracted justifications from different methods. The justifications are shown in random order for each comparison. In this example, our method corresponds to the third model. . . . .	125
D.3	Annotation platform for judging the quality of generated justifications from different methods, on four dimensions. The justifications are shown in random order for each comparison. In this example, our method corresponds to the second model. . . . .	126



# List of Tables

2.1	Statistics of the multi-aspect review datasets. Both datasets have high correlations between aspects. . . . .	19
2.2	Performance related to human evaluation, showing the precision of the selected words for each aspect of the <i>Beer</i> dataset. The percentage indicates the number of selected words of the reviews. MTM infers rationales that better correlate with human judgements. . . . .	22
2.3	Performance on automatic evaluation, showing the average topic coherence (NPMI) across different top- $N$ words for each dataset. We treat each aspect $a_i$ as a topic and use the masks/attentions to compute $P(w a_i)$ . MTM produces more coherent rationales. . . . .	23
2.4	Performance of the multi-aspect sentiment classification task for the <i>Beer</i> (top) and <i>Hotel</i> (bottom) datasets. MTM obtains the best F1 scores and requires fewer parameters. . . . .	24
2.5	Statistics of the multi-aspect review datasets. <i>Beer</i> and <i>Hotel</i> represent real-world beer and hotel reviews, respectively. <i>Decorrelated Beer</i> is a subset of the <i>Beer</i> dataset with a low-correlation assumption between aspect ratings, leading to a more straightforward and unrealistic dataset. . . . .	27
2.6	Performance of the multi-aspect sentiment classification task for the <b>decorrelated</b> <i>Beer</i> dataset. MTM still outperforms baselines in the unrealistic decorrelated settings. . . . .	27
3.1	Statistics of the review datasets. . . . .	40
3.2	Accuracy and objective performance of rationales in automatic evaluation for the Amazon dataset. ConRAT infers rationales that align with factual and counterfactual ones. . . . .	41
3.3	Objective performance of rationales for the multi-aspect beer reviews. ConRAT only uses the overall label and ignores the other aspect labels. All baselines are trained separately on each aspect rating. <b>Bold</b> and <u>underline</u> denote the best and second-best results, respectively. ConRAT is competitive with the fully-supervised baselines. . . . .	42
3.4	Impact of the number of concepts in ConRAT on the objective performance for the beer reviews. . . . .	45

## List of Tables

---

3.5	Ablation study of ConRAT with five concepts. The results confirm our intuition and that our regularizers enforce the interpretable desiderata. . . . .	45
4.1	Descriptive statistics of the datasets. . . . .	59
4.2	Human evaluation of explanations in terms of the win rate and the <b>best-worst</b> scaling. A score significantly different than Markers (post hoc Tukey HSD test) is denoted by ** for $p < 0.001$ . Human annotators significantly prefer Markers as explanations. . . . .	61
4.3	Generated justifications on automatic evaluation using language generation metrics. . . . .	62
4.4	Human evaluation of justifications in terms of best-worst scaling for <b>Overall</b> , <b>Fluency</b> , <b>Informativenss</b> , and <b>Relevance</b> . Most scores are significantly different than T-RECS (post hoc Tukey HSD test) with $p < 0.002$ . † denotes a nonsignificant score. Explanations generated by T-RECS are largely preferred. . . . .	62
4.5	Comparisons of the extracted justifications from different models for two hotels on the hotel dataset. Colors denote aspects while underline denotes EDUs classified as “good” justifications. . . . .	63
4.6	Comparisons of the extracted justifications from different models for two beers on the beer dataset. Colors denote aspects while underline denotes EDUs classified as “good” justifications. . . . .	64
4.7	Keyphrase explanation quality using raking metrics at $N = 10$ . . . . .	65
4.8	Performance of the rating prediction. Kendall’s $\tau$ denotes the correlation between the users’ own rankings and the models’ ones. T-RECS consistently produces better recommendation. . . . .	66
4.9	Performance of the Top- $N$ recommendation. . . . .	68
5.1	Descriptive statistics of the datasets. Coverage shows the ratio of reviews having at least one of the selected keyphrases (KPs) . . . . .	86
5.2	Some keyphrases mined from the reviews. We manually grouped them by types for a better understanding. . . . .	87
5.3	Top- $N$ recommendation results of all datasets. <b>Bold</b> and <u>underline</u> denote the best and second-best results, respectively. We omit the error bars because the 95% confidence interval is in 4 <sup>th</sup> digit. . . . .	89
5.4	Top- $K$ keyphrase explanation quality of all datasets. <b>Bold</b> and <u>underline</u> denote the best and second-best results, respectively. We omit the error bars because the 95% confidence interval is in 4 <sup>th</sup> digit. . . . .	90
A.1	Objective performance of rationales for the multi-aspect beer reviews with the concept length set to five. ConRAT only uses the overall rating and does not have access to the other aspect labels. All baselines are trained separately on each aspect label. <b>Bold</b> and <u>underline</u> denote the best and second-best results, respectively. . . . .	107



A.2	Performance of the generated personalized justifications on automatic evaluation. . . . .	109
A.3	Performance of personalized keyphrase explanation quality. . . . .	110
B.1	Some keyphrases mined from the inferred <i>markers</i> . We grouped them by aspect for a better understanding. . . . .	112
C.1	Top ten words for each aspect from the <i>Beer</i> dataset, learned by various models. <b>Red</b> denotes intruders according to two annotators. Found words are generally noisier due to the high correlation between <i>Taste</i> and other aspects. However, <i>MTM</i> provides better results than other methods. . . . .	114
C.2	Top ten words for each aspect from the <i>Hotel</i> dataset, learned by various models. <b>Red</b> denotes intruders according to human annotators. Besides <i>SAM</i> , all methods find similar words for most aspects except the aspect <i>Value</i> . The top words of <i>MTM</i> do not contain any intruder. . . . .	115
C.3	Top ten words for each aspect from the <i>decorrelated Beer</i> dataset, learned by various models. <b>Red</b> denotes intruders according to two annotators. For the three aspects, <i>MTM</i> has only one word considered as an intruder, followed by <i>MASA</i> with <i>SAM</i> (two) and <i>MAA</i> (six). . . . .	115
F.1	Best hyperparameter setting for each model. The top table refers to Section 5.5.3 and the bottom one to Section 5.5.4. . . . .	133



# 1 Introduction

Thanks to the era of big data and modern technology, the past decade has witnessed unprecedented progress in artificial intelligence (AI) and machine learning (ML). AI-based decisions now pervade our daily lives and offer a wide range of benefits. In many domains, including healthcare, criminal justice, job hiring, and education, automated predictions have a tangible impact on the final decision (Citron and Pasquale, 2014; Caruana et al., 2015; Barocas et al., 2019; Vyas et al., 2021). As research has shown, however, modern systems sometimes have unintentional and undesirable biases (e.g., Buolamwini and Gebru, 2018; Angwin et al., 2016; Pariser, 2011). Therefore, it is crucial to provide the underlying reasons for automated decisions.

In recent years, new initiatives and regulations have been proposed. In 2016, for example, the European Parliament adopted a set of comprehensive regulations for the collection, storage, and use of personal information: the European General Data Protection Regulation (GDPR)<sup>1</sup> (Goodman and Flaxman, 2017). Around the same time, the U.S. Defense Advanced Research Projects Agency (DARPA) created an explainable artificial intelligence program that also emphasizes the importance of human-computer interaction for ML systems (Gunning, 2016). Both of these initiatives stress a social “right to explanation”: users can ask for an explanation of any algorithmic decision or instance of profiling associated with them. Furthermore, prior studies have shown that explanations improve overall system transparency (e.g., Tintarev and Masthoff, 2015; Sinha and Swearingen, 2002) and trustworthiness (e.g., Zhang and Curley, 2018; Kunkel et al., 2018), increasing at an accelerating pace the variety of applications (Miller, 2019).

Nevertheless, current AI and ML systems are far from perfect and do make mistakes. Explainable systems are also concerned, because they can generate faulty, unfaithful, or fragile explanations independently of the correctness of the predictions. Dietvorst et al. (2015); Castelo et al. (2019); Jussupow et al. (2020); Smith-Renner et al. (2020) showed that people have less confidence in the models when seeing them err for some tasks, even when the

---

<sup>1</sup><https://eur-lex.europa.eu/eli/reg/2016/679/2016-05-04>.

machine performs better than humans. This algorithmic aversion can be overcome either by

1. letting users modify (even slightly) the models (Dietvorst et al., 2018) or
2. explaining how the models induce their predictions (Yeomans et al., 2019; Smith-Renner et al., 2020).

These strategies align with the view of Shneiderman et al. (2016): highlighting the need for users to understand better underlying computational processes in search, recommender, and other algorithms and the need to give to users the potential to control their actions better.

Interpretability for natural language processing tasks is a large, fast-growing area of research. It includes text classification (e.g., sentiment analysis, risk prediction for diabetes, anemia, and breast cancer), natural language inference, reading comprehension, and question answering (Bastings et al., 2019; DeYoung et al., 2020; Wiegrefe and Marasovic, 2021). Because more data are available than ever before, it has become increasingly important to explain automated predictions. In recent years, selective rationalization has become popular (Lei et al., 2016; Bastings et al., 2019; DeYoung et al., 2020; Yu et al., 2021). The key idea of selective rationalization is to explain complex neural networks by selecting relevant text snippets from the input text – called a rationale – that suffice on their own to yield the same prediction as the original outcome. Rationalization can be naturally extended to different types of inputs, like images. Moreover, in practice, a rationale can potentially be understood and verified against domain knowledge (Lei et al., 2016; Yala et al., 2019; Chang et al., 2019). However, rationales are limited by their single dimensionality: current rationalization models strive for one overall selection to explain the outcome. However, useful rationales can be multi-faceted, involving support for different outcomes with different degrees (Musat and Faltings, 2015; Chang et al., 2019).

Therefore, it is important to provide fine-grained explanations to users. Recommendation systems represent an impactful real-life application in which interactions are naturally present. There is not only a practical need to better model users and improve the recommendations themselves but also a growing demand for explainability<sup>2</sup>. Finally, many data are publicly available, and they contain user feedback and often user-written reviews; this facilitates research and development that targets new, data-driven techniques for explainable recommender systems.

In an increasingly digital world, recommender systems are an inevitable part of virtually everyone's daily digital routine. They influence how we perceive our environment, from media content to human relationships. In response to the continuously growing quantity of information, products, and choices, modern recommender systems suggest content tailored to users' interests. Most importantly, users deliver feedback and often write reviews about the products they consume. Research has shown that 1. users offer written opinions about

---

<sup>2</sup>For example, since July 2021, Google Search has partly explained to users why it found the results it shows them. [www.reuters.com/technology/google-is-starting-tell-you-how-it-found-search-results-2021-07-22/](https://www.reuters.com/technology/google-is-starting-tell-you-how-it-found-search-results-2021-07-22/).

---

the topics they care about (Zhang et al., 2014; Musat and Faltings, 2015) and 2. human explanations for high-level decisions are often based on key concepts (Armstrong et al., 1983; Tenenbaum, 1999). These observations underline that rationales are naturally present in user-written texts, which will play a key role in 1. the understanding of users' preferences and 2. better building their profiles to improve subsequent personalized recommendations. Finally, several studies (Konstan and Riedl, 2012; Knijnenburg et al., 2012) have emphasized that personalization reduces search effort and makes the user experience more comfortable and engaging. Furthermore, it plays a crucial role in many e-commerce websites, such as Amazon, Twitter, YouTube, and Netflix (Jannach and Jugovac, 2019). For example, in 2012 35% of purchases on Amazon and 75% of what users watch on Netflix were influenced by their recommender systems (MacKenzie et al., 2013).

Although modern recommender systems accurately capture users' preferences and achieve high performance, they offer little transparency regarding their inner workings. Thus, their performance comes at the cost of an increased complexity that makes them seem like black boxes to end users, which may result in distrust or rejection of the recommendations themselves (Herlocker et al., 2000; Tintarev and Masthoff, 2015). It has been shown that providing explanations along with item recommendations

1. enables users to understand why a particular item has been suggested and hence to make better decisions (Chang et al., 2016; Bellini et al., 2018),
2. increases the system's overall transparency for users (Sinha and Swearingen, 2002; Tintarev and Masthoff, 2015), and
3. improves users' perceived trustworthiness (Pu and Chen, 2006; Zhang and Curley, 2018; Kunkel et al., 2018).

Although it is important to provide explanations for recommendation systems, one obstacle to doing so involves **explanation generation**: how do we generate useful explanations? Multiple approaches to producing simple explanations have been proposed: aspect words (McAuley and Leskovec, 2013; Wu et al., 2019a), template sentences (Zhang et al., 2014), and user-item similarity (Herlocker et al., 2000). However, Kunkel et al. (2019); Chang et al. (2016); Wilkinson et al. (2021) showed that not all explanations are equivalent: highly personalized justifications<sup>3</sup> using natural language lead to substantial improvements in perceived recommendation quality and trustworthiness compared to more straightforward explanations. Nevertheless, generating useful and meaningful explanations remains a difficult task due to the lack of ground-truth datasets specifying what "good" justifications are and indicating whether they are multi-faceted.

An essential advantage of explanations is that they provide a basis for feedback. If users understand what has generated the suggestions, they can refine the recommendations by

---

<sup>3</sup>We use the terms "explanations", "rationales", and "justifications" interchangeably.

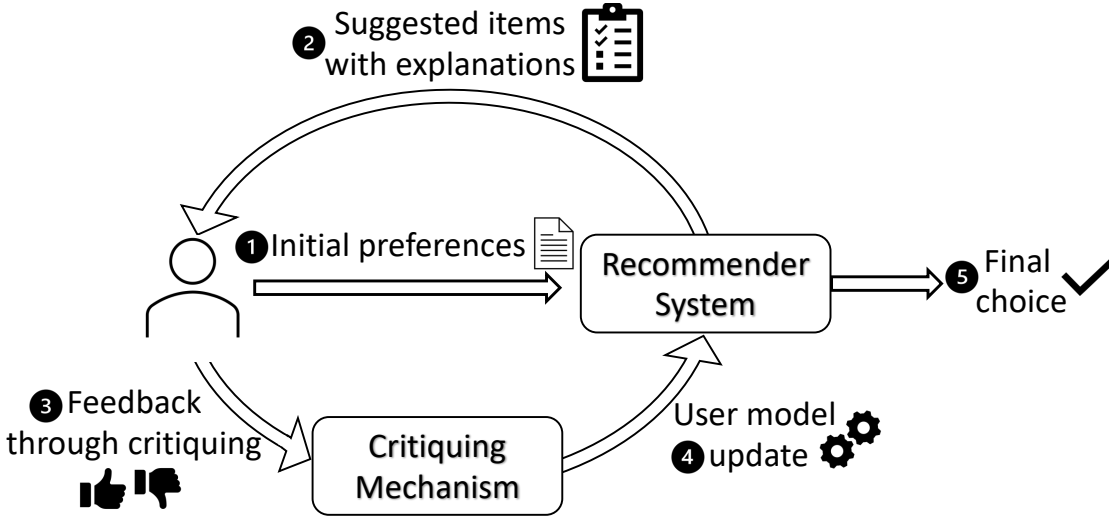


Figure 1.1: The cycle of conversational critiquing. First, the user informs the recommender system of his or her initial preferences. Then, the model returns a list of recommended items with their respective explanations. The user can interact with the explanations and critique them. This process (2 - 4) repeats until the user accepts the recommendation and ceases to provide critiques.

interacting directly with the explanations (Lyu et al., 2021). Therefore, the second challenge involves **critiquing** – a conversational recommendation method that incrementally adapts recommendations in response to user preferences (Chen and Pu, 2012). This iterative (i.e., multi-step) process continues until the user accepts the recommendation and ceases to provide critiques. Figure 1.1 illustrates the overall interaction cycle. It has been shown that critiquing has a substantive impact on the user’s experience and improves the user’s decision accuracy (Pu and Chen, 2005; Viappiani et al., 2006, 2008). Another benefit of critiquing is that it enables the system to correct and improve an incomplete or inaccurate model of the user’s preferences (Chen and Pu, 2007b).

Historically, most work on critiquing (e.g., Faltings et al., 2004a; Viappiani et al., 2006, 2008; Chen and Pu, 2007a,b) has assumed a fixed set of discrete and domain-dependent attributes along with explicit constraint- and utility-based methods. However, a major limitation of direct filtering-based interactions is assuming a fixed set of known attributes. Another drawback is their lack of effectiveness for domains with expressive feature sets: critiques might not generalize to other items with correlated (hidden) attributes (Wu et al., 2019a). Recently, few studies (Wu et al., 2019a; Luo et al., 2020a,b; Li et al., 2020) have assumed that item properties are represented as keyphrases sourced from subjective user reviews. However, it is not trivial to co-embed item critiques with the latent user preferences. These limitations hinder the pace at which companies are deploying explainable recommender systems.

## 1.1 Dissertation Structure and Contributions

This dissertation addresses the two aforementioned challenges: explanation generation and multi-step critiquing. More specifically, our approach first extracts textual explanations from natural language texts using novel multi-aspect rationales. Then, our method tailors and leverages textual explanations for *explainable* recommendation systems. Our critiquing methods enable explanations to be actionable, which in turn enables the recommenders to become conversational. Our framework is generalizable beyond recommendation systems and other domains, such as medicine, law, finance, or cooking. Examples include medical-report generation from radiography, diagnostic and procedural coding of clinical notes, the removal of biases and sensitive information from documents, and the rewriting of cooking recipes to satisfy dietary restrictions. Finally, the input data are not restricted to text.

Part I of this dissertation presents two approaches to extracting high-quality, personalized, multi-faceted justifications from text documents under the paradigm of selective rationalization. A unique advantage of rational models over many other interpretable methods (e.g., Bahdanau et al., 2015; Ribeiro et al., 2016; Serrano and Smith, 2019; Li et al., 2016b; Xu et al., 2015b; Chen et al., 2019a) is that unselected words are guaranteed not to contribute to the final prediction. Rationales are faithful (i.e., meaningfully influencing the model’s predictions) and can also be considered high-quality compressed information of a data sample with respect to the downstream task. This opens new doors for applications, such as user profiling, summarization, recommendation, and many others. In the two proposed methods, we infer multi-dimensional rationales and emphasize that a single overall selection does not accurately capture the multi-faceted nature of useful rationales. Moreover, we demonstrate the efficiency and scalability of our two approaches on massive corpora.

Based on the high-level idea of rationalization, Part II uses the induced multi-dimensional rationales – which reflect users’ ratings and subratings – to construct large personalized recommendation justification datasets and train models to infer personalized explanations alongside recommendations. The justifications are generated in an abstractive fashion and are intended to reflect the user’s preferences for each recommended item. We show that human users significantly prefer our explanations over those produced by state-of-the-art techniques. Moreover, using language-based feedback, the user can then interact with these explanations to refine the recommendation, enabling a type of reasoning from the machine side. By explicitly critiquing certain aspects of the items, the recommender can integrate these constraints for subsequent recommendations. We propose two approaches with different properties that achieve good performance in adapting to the preferences expressed in multi-step critiquing and generate consistent explanations without the cost of reduced recommendation performance. Finally, we introduce multiple model-agnostic interfaces to test such systems on an actual use case, and we highlight some limitations of these approaches in the hope that the research community will find ways to overcome them.

The two major parts of this dissertation address the challenges outlined above: selective

rationalization (i.e., explanation generation) and developing a conversational explainable recommendation system (i.e., explanation generation and critiquing). Both parts propose several innovative techniques along with their applications. Our dissertation’s main contributions are as follows:

1. **From single-dimensional to multi-dimensional rationales.** We first focus on selective rationalization for explaining the outcomes corresponding to *multiple* labels. We take the best of the attention and rationale methods and propose the Multi-Target Masker (MTM), which addresses their limitations by replacing the hard binary mask with a soft multi-dimensional mask (one for each target); it is learned in an unsupervised and multi-task learning manner while jointly predicting all the target variables. To the best of our knowledge, this study is the first to neither assume any prior on the data distribution nor train, tune, or maintain multiple models, one for each label to explain. Moreover, our experiments show that our proposed model significantly outperforms attention and strong baselines in terms of mask precision, coherence, and aspect rating performance. Thus, interpretability no longer comes at the cost of reduced performance.
2. **Generalizing multi-dimensional rationalization through latent concepts.** Our second contribution involves a more challenging and realistic setting than the first, but one that has the same advantages. We aim to learn the same multi-dimensional rationales while observing only *one* label. We present the Concept-based Rationalizer (ConRAT), a novel rationalization scheme that first identifies a set of concepts in a document and then decides which ones are currently described (binary selection). The model explains the prediction with a linear aggregation of concepts. In addition, we design two regularizers that guide the model to induce interpretable concepts and propose two optional techniques, knowledge distillation and concept pruning, in order to boost the performance further. Our method significantly outperforms strong *supervised* baseline models in terms of predictive and explanation performance. Finally, the concepts generated align with human rationalization and are preferred over rationales inferred by state-of-the-art methods trained on each aspect label independently.
3. **Generating large-scale personalized justification datasets for recommendation.** We introduce a data-driven and scalable pipeline, based on the previous contributions, that extracts high-quality and personalized justifications from users’ reviews but that is also applicable to other types of documents. The extracted text snippets reflect user ratings and subratings. Applied in the area of recommendation, we show that human users prefer our justifications more than 80% of the time over those produced by state-of-the-art techniques.
4. **Interacting with explanations through critiquing.** We examine the problem of recommendation with actionable fine-grained explanations. We propose a transformer-based model (T-RECS) that explains a rating by first inferring a set of keyphrases describing the intersection between the profiles of a user and an item. Conditioned on the keyphrases,



the model generates a synthetic personalized justification. We then leverage these explanations in a new unsupervised method for single-step and multi-step critiquing. Our system achieves good performance in adapting to the preferences expressed in multi-step critiquing and generates consistent explanations. In the continuation, we introduce four different web interfaces that help users make decisions and find their ideal item. Because our application is model-agnostic (for both recommender systems and critiquing mechanisms), it allows great flexibility and further extensions. Our interfaces are, above all, a useful tool for research on recommendation with critiquing and for highlighting some limitations of these approaches.

5. **Making multi-step critiquing faster and more efficient with self-supervision.** Finally, we address the current recommender-system models' inherent trade-off between the efficiency, recommendation, explanation, and critiquing performance. We present a novel multimodal variational autoencoder (M&Ms-VAE), trained under a weak supervision scheme to simulate both fully and partially observed variables. It is computationally efficient and handles critiques up to 25.6x faster than existing models. Last but not least, it does not suffer from the cold-start problem, thanks to our multimodal modeling and training scheme.

The five contributions outlined above are described in detail in the next four chapters (contributions 3 and 4 are presented together). Each chapter corresponds to one or more publications. All the chapters begin with a preface, consisting of a summary of the work and, when applicable, a detailed presentation of their individual contributions using the CRediT framework (Brand et al., 2015). The final chapter offers concluding remarks and discusses possible directions for future research.



# **Extracting Multi-Faceted Explanations from Text Documents**

**Part I**



## 2 Multi-Dimensional Rationalization

### 2.1 Preface

**Contribution and Sources.** This chapter is largely based on Antognini et al. (2021b). The detailed individual contributions are listed below using the CRediT taxonomy (Brand et al., 2015) (terms are selected as applicable).

Diego Antognini (author): Conceptualization, Methodology, Software, Validation, Investigation, Formal Analysis, Writing – Original Draft, Writing – Review & Editing.

Claudiu Musat: Writing – Review & Editing (supporting), Supervision (supporting).

Boi Faltings: Writing – Review & Editing, Administration, Supervision.

**Summary.** Automated predictions require explanations to be interpretable by humans. Past work used attention and rationale mechanisms to find words that predict the target variable of a document. Often though, they result in a tradeoff between noisy explanations or a drop in accuracy. Furthermore, rationale methods cannot capture the multi-faceted nature of justifications for multiple targets, because of the non-probabilistic nature of the mask. In this chapter, we propose the Multi-Target Masker (MTM) to address these shortcomings. The novelty lies in the soft multi-dimensional mask that models a relevance probability distribution over the set of target variables to handle ambiguities and remove the low-correlation assumption in the data. Additionally, two regularizers guide MTM to induce long, meaningful explanations. We evaluate MTM on two datasets and show, using standard metrics and human annotations, that the resulting masks are more accurate and coherent than those generated by the state-of-the-art methods. Moreover, MTM is the first to also achieve the highest F1 scores for all the target variables simultaneously while training a single model.

### 2.2 Introduction

Neural models have become the standard for natural language processing tasks. Despite the large performance gains achieved by these complex models, they offer little transparency about their inner workings. Thus, their performance comes at the cost of interpretability, limiting their practical utility. Integrating interpretability into a model would supply reasoning for the prediction, increasing its utility.

Perhaps the simplest means of explaining predictions of complex models is by selecting relevant input features. Prior work includes various methods to find relevant words in the text input to predict the target variable of a document. Attention mechanisms (Bahdanau et al., 2015; Luong et al., 2015) model the word selection by a conditional importance distribution over the inputs, used as explanations to produce a weighted context vector for downstream modules. However, their reliability has been questioned (Jain and Wallace, 2019; Pruthi et al., 2020). Another line of research includes rationale generation methods (Lundberg and Lee, 2017; Li et al., 2016b; Lei et al., 2016). If the selected text input features are short and concise – called a rationale or mask – and suffice on their own to yield the prediction, it can potentially be understood and verified against domain knowledge (Lei et al., 2016; Chang et al., 2019). Specifically, these rationale generation methods have been recently proposed to provide such explanations alongside the prediction. Ideally, a good rationale should yield the same or higher performance as using the full input.

The key motivation of this chapter arises from the limitations of existing methods. First, the attention mechanisms induce an importance distribution over the inputs, but the resulting explanation consists of many short and noisy word sequences (Figure 2.1). In addition, the rationale generation methods produce coherent explanations, but the rationales are based on a binary selection of words, leading to the following shortcomings:

1. they explain only one target variable,
2. they make a priori assumptions about the data, and
3. they make it difficult to capture ambiguities in the text.

Regarding the first shortcoming, rationales can be multi-faceted by definition and involve support for different outcomes. If that is the case, one has to train, tune, and maintain one model per target variable, which is impractical. For the second, current models are prone to pick up spurious correlations between the input features and the output. Therefore, one has to ensure that the data have low correlations among the target variables, although this may not reflect the real distribution of the data. Finally, regarding the last shortcoming, a strict assignment of words as rationales might lead to ambiguities that are difficult to capture. For example, in an hotel review that states “*The room was large, clean, and close to the beach.*”, the word “*room*” refers to the aspects *Room*, *Cleanliness*, and *Location*. All these limitations are implicitly related due to the non-probabilistic nature of the mask. For further illustrations, see Figure 2.4 and the Appendix C.2.



We show that it can be a benefit to:

1. guide the model to focus on different parts of the input text,
2. capture ambiguities of words belonging to multiple aspects, and
3. further improve the sentiment prediction for all the aspects.

Therefore, interpretability does not come at a cost in our paradigm.

## 2.3 Related Work

### 2.3.1 Interpretability

Developing interpretable models is of considerable interest to the broader research community; this is even more pronounced with neural models (Kim et al., 2015; Doshi-Velez and Kim, 2017). There has been much work with a multitude of approaches in the areas of analyzing and visualizing state activation (Karpathy et al., 2015; Li et al., 2016a; Montavon et al., 2018), attention weights (Jain and Wallace, 2019; Serrano and Smith, 2019; Pruthi et al., 2020; Clark et al., 2019), and learned sparse and interpretable word vectors (Faruqui et al., 2015b,a; Herbelot and Vecchi, 2015). Other works interpret black box models by locally fitting interpretable models (Ribeiro et al., 2016; Lundberg and Lee, 2017). Li et al. (2016b) proposed erasing various parts of the input text using reinforcement learning to interpret the decisions. However, this line of research aims at providing post-hoc explanations of an already-trained model. Our work differs from these approaches in terms of what is meant by an explanation and its computation. We defined an explanation as one or multiple text snippets that – as a substitute of the input text – are sufficient for the predictions.

### 2.3.2 Multi-Aspect Sentiment Classification

Multi-aspect sentiment classification is sometimes seen as a sub-problem of sentiment analysis. McAuley et al. (2012); Pappas and Popescu-Belis (2014) employed heuristic-based methods or topic models. Neural models achieve significant improvements with less feature engineering. Yin et al. (2017) built a hierarchical attention model with aspect representations by using a set of manually defined topics. Li et al. (2018a) extended this chapter with user attention and additional features such as overall rating, aspect, and user embeddings. The disadvantage of these methods is their limited interpretability because they rely on many features in addition to the review text.

### 2.3.3 Attention-based Models

Attention models (Vaswani et al., 2017; Yang et al., 2016; Lin et al., 2017) have been shown to improve prediction accuracy, visualization, and interpretability. The most popular and



widely used attention mechanism is soft attention (Bahdanau et al., 2015), rather than hard attention (Luong et al., 2015) or sparse ones (Martins and Astudillo, 2016). According to various studies (Jain and Wallace, 2019; Serrano and Smith, 2019; Pruthi et al., 2020), standard attention modules noisily predict input importance; the weights cannot provide safe and meaningful explanations. Moreover, Pruthi et al. (2020) showed that standard attention modules can fool people into thinking that predictions from a model biased against gender minorities do not rely on the gender. Our approach differs in two ways from attention mechanisms. First, the loss includes two regularizers to favor long word sequences for interpretability. Second, the normalization is not done over the sequence length but over the target set for each word; each has a relevance probability distribution over the set of target variables.

### 2.3.4 Rationale Models

The idea of using human rationales during training has been explored in Zhang et al. (2016); Bao et al. (2018); DeYoung et al. (2020). Although they have been shown to be beneficial, they are costly to collect and might vary across annotators. In our work, no annotation is needed.

One of the first rationale generation methods was introduced by Lei et al. (2016) in which a generator masks the input text fed to the classifier. This framework is a cooperative game that selects rationales to accurately predict the label by maximizing the mutual information (Chen et al., 2018b). Yu et al. (2019) proposed conditioning the generator based on the predicted label from a classifier reading the whole input, although it slightly underperformed when compared to the original model (Chang et al., 2020). Chang et al. (2019) presented a variant that generated rationales to perform counterfactual reasoning. Finally, Chang et al. (2020) proposed a generator that can decrease spurious correlations in which the selective rationale consists of an extracted chunk of a pre-specified length, an easier variant than the original one that generated the rationale. In all, these models are trained to generate a hard binary mask as a rationale to explain the prediction of a target variable, and the method requires as many models to train as variables to explain. Moreover, they rely on the assumption that the data have low internal correlations.

In contrast, our model addresses these drawbacks by jointly predicting the rationales of all the target variables (even in the case of highly correlated data) by generating a soft multi-dimensional mask. The probabilistic nature of the masks can handle ambiguities in the induced rationales. In Chapter 4, we show how to use the induced rationales to generate personalized explanations for recommendation and how human users significantly prefer these over those produced by state-of-the-art models.

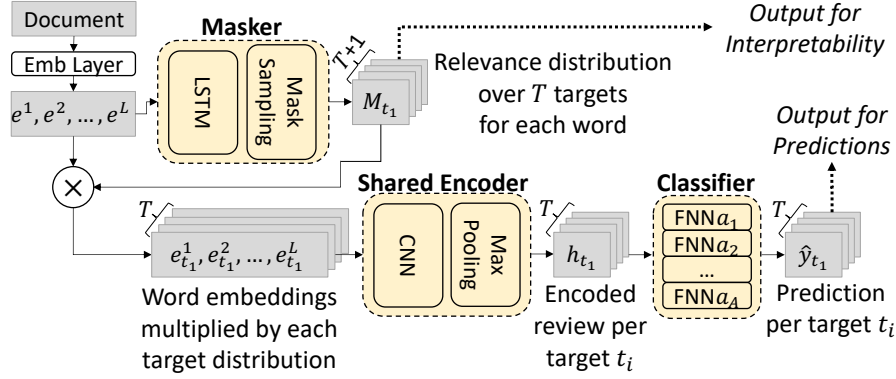


Figure 2.2: The proposed Multi-Target Masker (MTM) model architecture to predict and explain  $T$  target variables.

## 2.4 MTM: A Multi-Target Masker

Let  $X$  be a random variable representing a document composed of  $L$  words  $(x^1, x^2, \dots, x^L)$ , and  $Y$  the target  $T$ -dimensional vector<sup>1</sup>. Our proposed model, called the Multi-Target Masker (MTM), is composed of three components: 1) a **masker** module that computes a probability distribution over the target set for each word, resulting in  $T + 1$  masks (including one for the irrelevant case); 2) an **encoder** that learns a representation of a document  $X$  conditioned on the induced masks; 3) a **classifier** that predicts the target variables. The overall model architecture is shown in Figure 2.2. Each module is interchangeable with other models.

### 2.4.1 Model Overview

#### Masker.

The masker first computes a hidden representation  $h^\ell$  for each word  $x^\ell$  in the input sequence, using their word embeddings  $e^1, e^2, \dots, e^L$ . Many sequence models could realize this task, such as recurrent, attention, or convolution networks. In our case, we chose a recurrent model to learn the dependencies between the words. Let  $t_i$  be the  $i^{th}$  target for  $i = 1, \dots, T$ , and  $t_0$  the irrelevant case, because many words are irrelevant to every target. We define the multi-dimensional mask  $\mathbf{M} \in \mathbb{R}^{(T+1) \times L}$  as the target relevance distribution  $M^\ell \in \mathbb{R}^{(T+1)}$  of each word  $x^\ell$  as follows:

$$P(\mathbf{M}|X) = \prod_{\ell=1}^L P(M^\ell|x^\ell) = \prod_{\ell=1}^L \prod_{i=0}^T P(m_{t_i}^\ell|x^\ell). \quad (2.1)$$

Because we have categorical distributions, we cannot directly sample  $P(M^\ell|x^\ell)$  and back-propagate the gradient through this discrete generation process. Instead, we model the variable  $m_{t_i}^\ell$  using the straight through gumbel-softmax (Jang et al., 2017; Maddison et al.,

<sup>1</sup>Our method is easily adapted for regression problems.

2017) to approximate sampling from a categorical distribution<sup>2</sup>. We model the parameters of each Gumbel-Softmax distribution  $M^\ell$  with a single-layer feed-forward neural network followed by applying a log softmax, which induces the log-probabilities of the  $\ell^{th}$  distribution:  $\omega_\ell = \log(\text{softmax}(Wh^\ell + b))$ .  $W$  and  $b$  are shared across all tokens so that the number of parameters stays constant with respect to the sequence length. We control the sharpness of the distributions with the temperature parameter  $\tau$ , which dictates the peakiness of the relevance distributions. In our case, we keep the temperature low to enforce the assumption that each word is relevant about one or two targets. Note that compared to attention mechanisms, the word importance is a probability distribution over the targets  $\sum_{i=0}^T P(m_{t_i}^\ell | x^\ell) = 1$  instead of a normalization over the sequence length  $\sum_{\ell=1}^L P(t^\ell | x^\ell) = 1$ .

Given a soft multi-dimensional mask  $\mathbf{M} \in \mathbb{R}^{(T+1) \times L}$  and the input  $X \in \mathbb{R}^L$ , we define each sub-mask  $M_{t_i} \in \mathbb{R}^L$  as follows:

$$M_{t_i} = P(m_{t_i}^1 | x^1), P(m_{t_i}^2 | x^2), \dots, P(m_{t_i}^L | x^L). \quad (2.2)$$

To integrate the word importance of the induced sub-masks  $M_{t_i}$  within the model, we weight the word embeddings by their importance towards a target variable  $t_i$ , such that  $E_{t_i} = E \odot M_{t_i} = e_1 \cdot P(m_{t_i}^1 | x^1), e_2 \cdot P(m_{t_i}^2 | x^2), \dots, e_L \cdot P(m_{t_i}^L | x^L)$ . Thereafter, each modified embedding  $E_{t_i}$  is fed into the encoder block. Note that  $E_{t_0}$  is ignored because  $M_{t_0}$  only serves to absorb probabilities of words that are insignificant<sup>3</sup>.

### Encoder and Classifier.

The encoder includes a convolutional network, followed by max-over-time pooling to obtain a fixed-length feature vector. We chose a convolutional network because it led to a smaller model, faster training, and performed empirically similarly to recurrent and attention models. It produces the fixed-size hidden representation  $h_{t_i}$  for each target  $t_i$ . To exploit commonalities and differences among the targets, we share the weights of the encoder for all  $E_{t_i}$ . Finally, the classifier block contains for each target variable  $t_i$  a two-layer feedforward neural network, followed by a softmax layer to predict the outcome  $\hat{y}_{t_i}$ .

### Extracting Rationales.

To explain the prediction  $\hat{y}_{t_i}$  of one target  $Y_{t_i}$ , we generate its rationale by selecting each word  $x^\ell$ , whose relevance towards  $t_i$  is the most likely:  $P(m_{t_i}^\ell | x^\ell) = \max_{j=0, \dots, T} P(m_{t_j}^\ell | x^\ell)$ . Then, we can interpret  $P(m_{t_i}^\ell | x^\ell)$  as the model confidence of  $x^\ell$  relevant to  $Y_{t_i}$ .

<sup>2</sup>We also experimented with the implicit reparameterization trick using a Dirichlet distribution (Figurnov et al., 2018) instead, but we did not obtain a significant improvement.

<sup>3</sup>if  $P(m_{t_0}^\ell | x^\ell) \approx 1.0$ , it implies  $\sum_{i=1}^T P(m_{t_i}^\ell | x^\ell) \approx 0$  and consequently,  $e_{t_i}^\ell \approx \tilde{0}$  for  $i = 0, \dots, T$ .

### 2.4.2 Enabling the Interpretability of Masks

The first objective to optimize is the prediction loss, represented as the cross-entropy between the true target label  $y_{t_i}$  and the prediction  $\hat{y}_{t_i}$  as follows:

$$\ell_{pred} = \sum_{i=1}^T \ell_{cross\_entropy}(y_{t_i}, \hat{y}_{t_i}). \quad (2.3)$$

However, training MTM to optimize  $\ell_{pred}$  will lead to meaningless sub-masks  $M_{t_i}$  because the model tends to focus on certain words. Consequently, we guide the model to produce long, meaningful word sequences, as shown in Figure 2.1. We propose two regularizers to control the number of selected words and encourage consecutive words to be relevant to the same target. For the first term, we calculate the probability  $p_{sel}$  of tagging a word as relevant to any target using the non-aspect mask  $m_{t_0}^\ell$  for each token  $x^\ell$  as follows:

$$p_{sel} = \frac{1}{L} \sum_{\ell=1}^L (1 - P(m_{t_0}^\ell | x^\ell)). \quad (2.4)$$

We then compute the cross-entropy with a prior hyperparameter  $\lambda_p$  to control the expected number of selected words among all target variables, which corresponds to the expectation of a binomial distribution ( $p_{sel}$ ). We minimize the difference between  $p_{sel}$  and  $\lambda_p$  as follows:

$$\ell_{sel} = \ell_{binary\_cross\_entropy}(p_{sel}, \lambda_p). \quad (2.5)$$

The second regularizer discourages the target transition of two consecutive words by minimizing the mean variation of their target distributions,  $M^\ell$  and  $M^{\ell-1}$ . We generalize the formulation of a hard binary selection as suggested by Lei et al. (2016) to a soft probabilistic multi-target selection as follows<sup>4</sup>:

$$p_{dis} = \frac{1}{L} \sum_{\ell=1}^L \frac{\|M^\ell - M^{\ell-1}\|_1}{A + 1}, \quad (2.6)$$

$$\ell_{cont} = \ell_{binary\_cross\_entropy}(p_{dis}, 0).$$

We train our Multi-Target Masker end-to-end and optimize the loss  $\ell_{MTM} = \ell_{pred} + \lambda_{sel} \cdot \ell_{sel} + \lambda_{cont} \cdot \ell_{cont}$ , where  $\lambda_{sel}$  and  $\lambda_{cont}$  control the impact of each constraint.

## 2.5 Experiments

We assess our model in two dimensions: the quality of the explanations and the predictive performance. Following previous work (Lei et al., 2016; Chang et al., 2020), we use sentiment analysis as a demonstration use case, but we extend it to the multi-aspect case. However, we are interested in learning rationales for every aspect at the same time without any prior

---

<sup>4</sup>Early experiments with other distance functions, such as the Kullback–Leibler divergence, produced inferior results.

Table 2.1: Statistics of the multi-aspect review datasets. Both datasets have high correlations between aspects.

Dataset	Beer	Hotel
Number of reviews	1,586,259	140,000
Average words per review	$147.1 \pm 79.7$	$188.3 \pm 50.0$
Average sentences per review	$10.3 \pm 5.4$	$10.4 \pm 4.4$
Number of Aspects	4	5
Avg./Max corr. between aspects	71.8%/73.4%	63.0%/86.5%

assumption on the data, where aspect ratings can be highly correlated. We first measure the quality of the induced rationales using human aspect sentence-level annotations and an automatic topic model evaluation method. In the second set of experiments, we evaluate MTM on the multi-aspect sentiment classification task in two different domains.

### 2.5.1 Experimental Details

The review encoder was either a bi-directional recurrent neural network using LSTM (Hochreiter and Schmidhuber, 1997) with 50 hidden units or a multi-channel text convolutional neural network, similar to Kim et al. (2015), with 3-, 5-, and 7-width filters and 50 feature maps per filter. Each aspect classifier is a two-layer feedforward neural network with a rectified linear unit activation function (Nair and Hinton, 2010). We used the 200-dimensional pretrained word embeddings of Lei et al. (2016) for beer reviews. For the hotel domain, we trained word2vec (Mikolov et al., 2013) on a large collection of hotel reviews (Antognini and Faltings, 2020b) with an embedding size of 300. We used a dropout (Srivastava et al., 2014) of 0.1, clipped the gradient norm at 1.0, added a L2-norm regularizer with a factor of  $10^{-6}$ , and trained using early stopping with a patience of three iterations. We used Adam (Kingma and Ba, 2015) with a learning rate of 0.001. The temperature  $\tau$  for the Gumbel-Softmax distributions was fixed at 0.8. The two regularizers and the prior of our model were  $\lambda_{sel} = 0.03$ ,  $\lambda_{cont} = 0.03$ , and  $\lambda_p = 0.15$  for the *Beer* dataset and  $\lambda_{sel} = 0.02$ ,  $\lambda_{cont} = 0.02$ , and  $\lambda_p = 0.10$  for the *Hotel* one. We ran all experiments for a maximum of 50 epochs with a batch-size of 256. We tuned all models on the dev set with 10 random search trials. For Lei et al. (2016), we used the code from the authors. For reproducibility purposes, we include additional details in Appendix F.1.

### 2.5.2 Datasets

McAuley et al. (2012) provided 1.5 million English beer reviews from BeerAdvocate. Each contains multiple sentences describing various beer aspects: *Appearance*, *Smell*, *Palate*, and *Taste*; users also provided a five-star rating for each aspect. To evaluate the robustness of the models across domains, we sampled 140,000 hotel reviews from Antognini and Faltings (2020b), that contains 50 million reviews from TripAdvisor. Each review contains a five-star rating for each aspect: *Service*, *Cleanliness*, *Value*, *Location*, and *Room*. The descriptive

statistics are shown in Table 2.1.

There are high correlations among the rating scores of different aspects in the same review (71.8% and 63.0% on average for the beer and hotel datasets, respectively). This makes it difficult to directly learn textual justifications for single-target rationale generation models (Chang et al., 2020, 2019; Lei et al., 2016). Prior work used separate decorrelated train sets for each aspect and excluded aspects with a high correlation, such as *Taste*, *Room*, and *Value*. However, these assumptions do not reflect the real data distribution. Therefore, we keep the original data (and thus can show that our model does not suffer from the high correlations). We binarize the problem as in previous work (Bao et al., 2018; Chang et al., 2020): ratings at three and above are labeled as positive and the rest as negative. We split the data into 80/10/10 for the train, validation, and test sets. Compared to the beer reviews, the hotel ones were longer, noisier, and less structured, as shown in Appendices C.2. Both datasets do not contain annotated rationales.

### 2.5.3 Baselines

We compare our Multi-Target Masker (*MTM*) with various baselines. We group them in three levels of interpretability:

- *None*. We cannot extract the input features the model used to make the predictions;
- *Coarse-grained*. We can observe what parts of the input a model used to discriminate all aspect sentiments without knowing what part corresponded to what aspect;
- *Fine-grained*. For each aspect, a model selects input features to make the prediction.

We first use a simple baseline, *SENT*, that reports the majority sentiment across the aspects, as the aspect ratings are highly correlated. Because this information is not available at testing, we trained a model to predict the majority sentiment of a review as suggested by Wang and Manning (2012). The second baseline we used is a shared encoder followed by  $T$  classifiers that we denote *BASE*. These models do not offer any interpretability. We extend it with a shared attention mechanism (Bahdanau et al., 2015) after the encoder, noted as *SAA* in our study, that provides a coarse-grained interpretability; for all aspects, *SAA* focuses on the same words in the input.

Our final goal is to achieve the best performance and provide fine-grained interpretability in order to visualize what sequences of words a model focuses on to predict the aspect sentiments. To this end, we include other baselines: two trained *separately* for each aspect (e.g., current rationale models) and two trained with a *multi-aspect* sentiment loss. For the first ones, we employ the well-known *NB-SVM* (Wang and Manning, 2012) for sentiment analysis tasks, and we then use the Single-Aspect Masker (*SAM*) (Lei et al., 2016), each trained separately for each aspect.

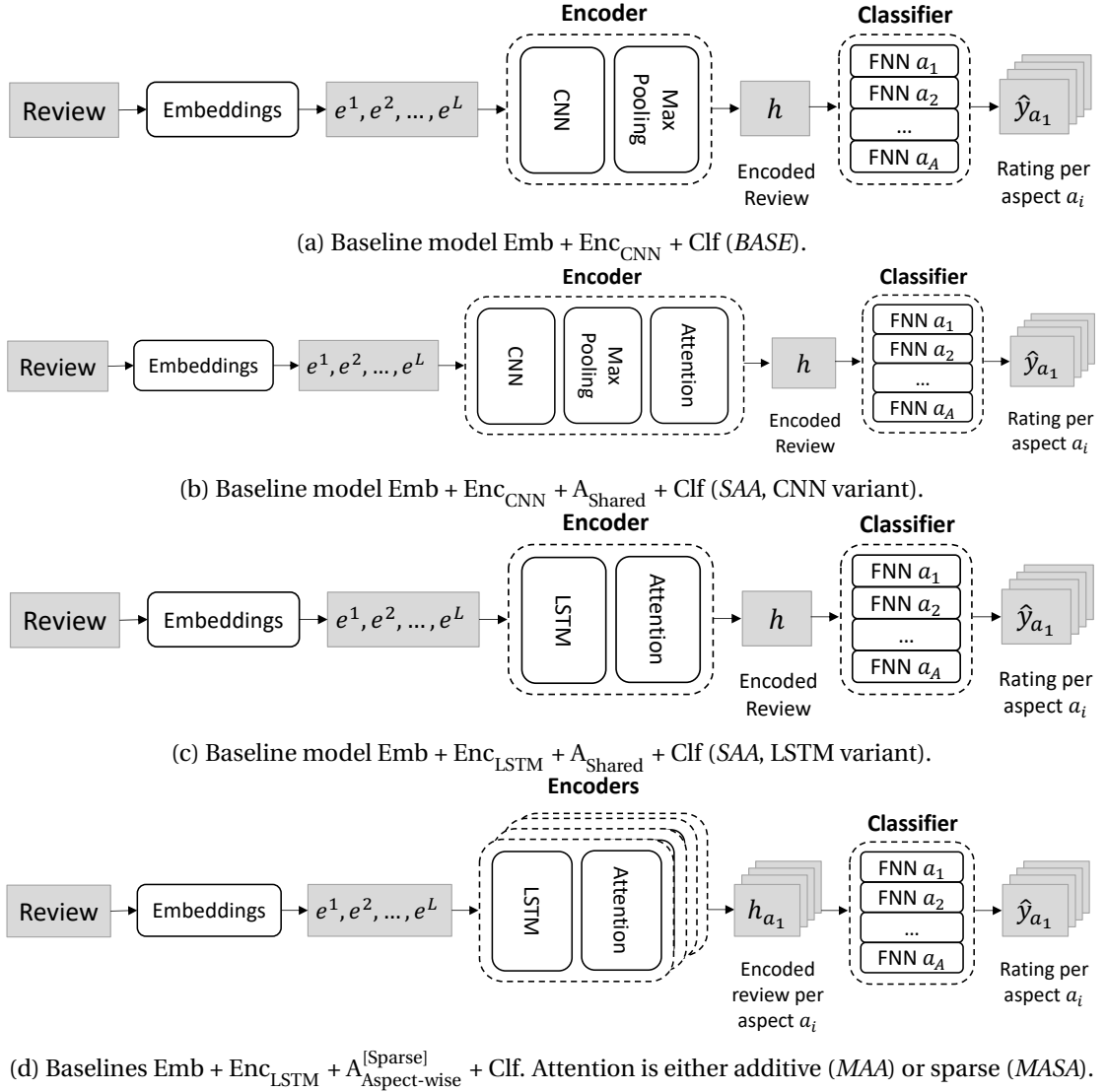


Figure 2.3: Baseline architectures.

The two last methods contain a separate encoder, attention mechanism, and classifier for each aspect. We utilize two types of attention mechanisms, additive (Bahdanau et al., 2015) and sparse (Martins and Astudillo, 2016), as sparsity in the attention has been shown to induce useful, interpretable representations. We call them Multi-Aspect Attentions (*MAA*) and Sparse-Attentions (*MASA*), respectively. Diagrams of the baselines can be found in Figure 2.3.

We demonstrate that the induced sub-masks  $M_{t_1}, \dots, M_{t_T}$  computed from *MTM* bring fine-grained interpretability and are meaningful for other models to improve performance. To do so, we extract and concatenate the masks to the word embeddings, resulting in contextualized embeddings (Peters et al., 2018), and train *BASE* with those. We call this variant *MTM<sup>C</sup>*, that is smaller and has faster inference than *MTM*.

Table 2.2: Performance related to human evaluation, showing the precision of the selected words for each aspect of the *Beer* dataset. The percentage indicates the number of selected words of the reviews. *MTM* infers rationales that better correlate with human judgements.

<b>Model</b>	<b>Precision / % Highlighted Words</b>		
	<i>Smell</i>	<i>Palate</i>	<i>Appearance</i>
NB-SVM*	21.6 / 7%	24.9 / 7%	38.3 / 13%
SAA*	88.4 / 7%	65.3 / 7%	80.6 / 13%
SAM*	95.1 / 7%	80.2 / 7%	96.3 / 14%
MASA	87.0 / 4%	42.8 / 5%	74.5 / 4%
MAA	51.3 / 7%	32.9 / 7%	44.9 / 14%
<b>MTM (Ours)</b>	<b>96.6 / 7%</b>	<b>81.7 / 7%</b>	<b>96.7 / 14%</b>

\* Model trained separately for each aspect.

## 2.6 Results

### 2.6.1 Multi-Rationale Interpretability

We first verify whether the inferred rationales of *MTM* are meaningful and interpretable, compared to the other models.

#### Precision.

Evaluating explanations that consist of coherent pieces of text is challenging because there is no gold standard for reviews. McAuley et al. (2012) have provided 994 beer reviews with sentence-level aspect annotations (although our model computes masks at a finer level). Each sentence was annotated with one aspect label, indicating what aspect that sentence covered. We evaluate the precision of the words selected by each model, as in Lei et al. (2016). We use trained models on the *Beer* dataset and extracted a similar number of selected words for a fair comparison. We also report the results of the models from Lei et al. (2016): *NB-SVM*, the Single-Aspect Attention and Masker (*SAA* and *SAM*, respectively); they use the separate decorrelated train sets for each aspect because they compute hard masks<sup>5</sup>.

Table 2.2 presents the precision of the masks and attentions computed on the sentence-level aspect annotations. We show that the generated sub-masks obtained with our Multi-Target Masker (*MTM*) correlates best with the human judgment. In comparison to *SAM*, the *MTM* model obtains significantly higher precision with an average of +1.13. Interestingly, *NB-SVM* and attention models (*SAA*, *MASA*, and *MAA*) perform poorly compared with the mask models, especially *MASA*, which focuses only on a couple of words due to the sparseness of the attention. In Section 2.6.3, we also analyze the impact of the length of the explanations.

<sup>5</sup>When trained on the original data, they performed significantly worse, showing the limitation in handling correlated variables.



Table 2.3: Performance on automatic evaluation, showing the average topic coherence (NPMI) across different top- $N$  words for each dataset. We treat each aspect  $a_i$  as a topic and use the masks/attentions to compute  $P(w|a_i)$ . MTM produces more coherent rationales.

	NPMI						
Model	$N = 5$	10	15	20	25	30	Mean <sup>†</sup>
<i>Beer</i>							
SAM <sup>*</sup>	0.046	0.120	0.129	0.243	0.308	0.396	0.207
MASA	0.020	0.082	0.130	0.168	0.234	0.263	0.150
MAA	0.064	<b>0.189</b>	0.255	0.273	0.332	0.401	0.252
<b>MTM</b>	<b>0.083</b>	0.187	<b>0.264</b>	<b>0.348</b>	<b>0.477</b>	<b>0.410</b>	<b>0.295</b>
<i>Hotel</i>							
SAM <sup>*</sup>	0.041	0.103	0.152	0.180	0.233	0.281	0.165
MASA	0.043	0.127	0.166	0.295	0.323	0.458	0.235
MAA	0.128	0.218	<b>0.352</b>	0.415	0.494	0.553	0.360
<b>MTM</b>	<b>0.134</b>	<b>0.251</b>	0.349	<b>0.496</b>	<b>0.641</b>	<b>0.724</b>	<b>0.432</b>

<sup>\*</sup> Model trained separately for each aspect.

<sup>†</sup> The metric that correlates best with human judgment (Lau and Baldwin, 2016).

### Semantic Coherence.

In addition to evaluating the rationales with human annotations, we compute their semantic interpretability. According to Aletras and Stevenson (2013); Lau et al. (2014), normalized point mutual information (NPMI) is a good metric for the qualitative evaluation of topics because it matches human judgment most closely. However, the top- $N$  topic words used for evaluation are often selected arbitrarily. To alleviate this problem, we followed Lau and Baldwin (2016). We compute the topic coherence over several cardinalities and report the results and average (see Appendix C.1); those authors claimed that the mean leads to a more stable and robust evaluation.

The results are shown in Table 2.3. We show that the computed masks by *MTM* lead to the highest mean NPMI and, on average, 20% superior results in both datasets, while only needing a single training. Our *MTM* model significantly outperforms *SAM* and the attention models (*MASA* and *MAA*) for  $N \geq 20$  and  $N = 5$ . For  $N = 10$  and  $N = 15$ , *MTM* obtains higher scores in two out of four cases (+.033 and +.009). For the other two, the difference was below .003. *SAM* obtains poor results in all cases and must be trained as many times as the number of aspects.

We analyzed the top words for each aspect by conducting a human evaluation to identify intruder words (i.e., words not matching the corresponding aspect). Generally, our model found better topic words: approximately 1.9 times fewer intruders than other methods for each aspect and each dataset. More details are available in Appendix C.1.

## Chapter 2. Multi-Dimensional Rationalization

Table 2.4: Performance of the multi-aspect sentiment classification task for the *Beer* (top) and *Hotel* (bottom) datasets. MTM obtains the best F1 scores and requires fewer parameters.

					F1 Scores					
Interp.		Model		Params	Macro	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	
Beer Reviews	None	SENT	Sentiment Majority	560 <i>k</i>	73.01	71.83	75.65	71.26	73.31	
		BASE	Emb <sub>200</sub> + Enc <sub>CNN</sub> + Clf	188 <i>k</i>	76.45	71.44	78.64	74.88	80.83	
	Coarse-grained	SAA	Emb <sub>200</sub> + Enc <sub>CNN</sub> + A <sub>Shared</sub> + Clf	226 <i>k</i>	77.06	73.44	78.68	75.79	80.32	
			Emb <sub>200</sub> + Enc <sub>LSTM</sub> + A <sub>Shared</sub> + Clf	219 <i>k</i>	78.03	74.25	79.53	75.76	82.57	
	Fine-grained	NB-SVM	(Wang and Manning, 2012)	4 · 560 <i>k</i>	72.11	72.03	74.95	68.11	73.35	
		SAM	(Lei et al., 2016)	4 · 644 <i>k</i>	76.62	72.93	77.94	75.70	79.91	
		MASA	Emb <sub>200</sub> + Enc <sub>LSTM</sub> + A <sub>Aspect-wise</sub> <sup>Sparse</sup> + Clf	611 <i>k</i>	77.62	72.75	79.62	75.81	82.28	
		MAA	Emb <sub>200</sub> + Enc <sub>LSTM</sub> + A <sub>Aspect-wise</sub> + Clf	611 <i>k</i>	78.50	74.58	79.84	77.06	82.53	
		MTM	Emb <sub>200</sub> + Masker + Enc <sub>CNN</sub> + Clf (Ours)	289 <i>k</i>	78.55	74.87	79.93	77.39	82.02	
		MTM <sup>C</sup>	Emb <sub>200+4</sub> + Enc <sub>CNN</sub> + Clf (Ours)	191 <i>k</i>	<b>78.94</b>	<b>75.02</b>	<b>80.17</b>	<b>77.86</b>	<b>82.71</b>	
					F1 Scores					
Interp.		Model		Params	Macro	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>
Hotel Reviews	None	SENT	Sentiment Majority	309 <i>k</i>	85.91	89.98	90.70	92.12	65.09	91.67
		BASE	Emb <sub>300</sub> + Enc <sub>CNN</sub> + Clf	263 <i>k</i>	90.30	92.91	93.55	94.12	76.65	94.29
	Coarse-grained	SAA	Emb <sub>300</sub> + Enc <sub>CNN</sub> + A <sub>Shared</sub> + Clf	301 <i>k</i>	90.12	92.73	93.55	93.76	76.40	94.17
			Emb <sub>300</sub> + Enc <sub>LSTM</sub> + A <sub>Shared</sub> + Clf	270 <i>k</i>	88.22	91.13	92.19	93.33	71.40	93.06
	Fine-grained	NB-SVM	(Wang and Manning, 2012)	5 · 309 <i>k</i>	87.17	90.04	90.77	92.30	71.27	91.46
		SAM	(Lei et al., 2016)	5 · 824 <i>k</i>	87.52	91.48	91.45	92.04	70.80	91.85
		MASA	Emb <sub>200</sub> + Enc <sub>LSTM</sub> + A <sub>Aspect-wise</sub> <sup>Sparse</sup> + Clf	1010 <i>k</i>	90.23	93.11	93.32	93.58	77.21	93.92
		MAA	Emb <sub>300</sub> + Enc <sub>LSTM</sub> + A <sub>Aspect-wise</sub> + Clf	1010 <i>k</i>	90.21	92.84	93.34	93.78	76.87	94.21
		MTM	Emb <sub>300</sub> + Masker + Enc <sub>CNN</sub> + Clf (Ours)	404 <i>k</i>	89.94	92.84	92.95	93.91	76.27	93.71
		MTM <sup>C</sup>	Emb <sub>300+5</sub> + Enc <sub>CNN</sub> + Clf (Ours)	267 <i>k</i>	<b>90.79</b>	<b>93.38</b>	<b>93.82</b>	<b>94.55</b>	<b>77.47</b>	<b>94.71</b>

### 2.6.2 Multi-Aspect Sentiment Classification

We showed that the inferred rationales of *MTM* were significantly more accurate and semantically coherent than those produced by the other models. Now, we inquire as to whether the masks could become a benefit rather than a cost in performance for the multi-aspect sentiment classification.

#### Beer Reviews.

We report the macro F1 and individual score for each aspect  $A_i$ . Table 2.4 (top) presents the results for the *Beer* dataset. The Multi-Target Masker (*MTM*) performs better on average than all the baselines and provided fine-grained interpretability. Moreover, *MTM* has two times

fewer parameters than the aspect-wise attention models.

The contextualized variant  $MTM^C$  achieves a macro F1 score absolute improvement of 0.44 and 2.49 compared to  $MTM$  and  $BASE$ , respectively. These results highlight that the inferred masks are meaningful to improve the performance while bringing fine-grained interpretability to  $BASE$ . It is 1.5 times smaller than  $MTM$  and has a faster inference.

$NB-SVM$ , which offers fine-grained interpretability and was trained separately for each aspect, significantly underperforms when compared to  $BASE$  and, surprisingly, to  $SENT$ . As shown in Table 2.1, the sentiment correlation between any pair of aspects of the *Beer* dataset is on average 71.8%. Therefore, by predicting the sentiment of one aspect correctly, it is likely that other aspects share the same polarity. We suspect that the linear model  $NB-SVM$  cannot capture the correlated relationships between aspects, unlike the non-linear (neural) models that have a higher capacity. The shared attention models perform better than  $BASE$  but provide only coarse-grained interpretability.  $SAM$  is outperformed by all the models except  $SENT$ ,  $BASE$ , and  $NB-SVM$ .

### Model Robustness - Hotel Reviews.

We check the robustness of our model on another domain. Table 2.4 (bottom) presents the results of the *Hotel* dataset. The contextualized variant  $MTM^C$  outperforms all other models significantly with a macro F1 score improvement of 0.49. Moreover, it achieves the best individual F1 score for each aspect  $A_i$ . This shows that the learned mask  $\mathbf{M}$  of  $MTM$  is again meaningful because it increases the performance and adds interpretability to  $BASE$ . Regarding  $MTM$ , we see that it performs slightly worse than the aspect-wise attention models  $MASA$  and  $MAA$  but has 2.5 times fewer parameters.

A visualization of a truncated hotel review with the extracted rationales and attentions is available in Figure 2.4. Not only do probabilistic masks enable higher performance, they better capture parts of reviews related to each aspect compared to other methods. More samples of beer and hotel reviews can be found in Appendix C.2.

To summarize, we have shown that the regularizers in  $MTM$  guide the model to produce high-quality masks as explanations while performing slightly better than the strong attention models in terms of prediction performance. However, we demonstrated that including the inferred masks into word embeddings and training a simpler model achieved the best performance across two datasets and at the same time, brought fine-grained interpretability. Finally,  $MTM$  supported high correlation among multiple target variables.

### Hard Mask versus Soft Masks.

$SAM$  is the neural model that obtained the lowest relative macro F1 score in the two datasets compared with  $MTM^C$ : a difference of  $-2.32$  and  $-3.27$  for the *Beer* and *Hotel* datasets, respec-



Figure 2.4: Induced rationales on a truncated hotel review, where shade colors represent the model confidence towards the aspects. *MTM* finds most of the crucial spans of words with a small amount of noise. *SAM* lacks coverage but identifies words where half are correct and the others ambiguous (represented with colored underlines).

tively. Both datasets have a high average correlation between the aspect ratings: 71.8% and 63.0%, respectively (see Table 2.1). Therefore, it makes it challenging for rationale models to learn the justifications of the aspect ratings directly. Following the observations of Lei et al. (2016); Chang et al. (2019, 2020), this highlights that single-target rationale models suffer from high correlations and require data to satisfy certain constraints, such as low correlations. In contrast, *MTM* does not require any particular assumption on the data.

We compare *MTM* in a setting where the aspect ratings were less correlated, although it does not reflect the real distribution of the aspect ratings. We employ the decorrelated subsets of

Table 2.5: Statistics of the multi-aspect review datasets. *Beer* and *Hotel* represent real-world beer and hotel reviews, respectively. *Decorrelated Beer* is a subset of the *Beer* dataset with a low-correlation assumption between aspect ratings, leading to a more straightforward and unrealistic dataset.

Dataset	Beer	Hotel	Decorrelated Beer
Number of reviews	1,586,259	140,000	280,000
Average word-length of review	$147.1 \pm 79.7$	$188.3 \pm 50.0$	$157.5 \pm 84.3$
Average sentence-length of review	$10.3 \pm 5.4$	$10.4 \pm 4.4$	$11.0 \pm 5.7$
Number of aspects	4	5	3
Average correlation between aspects	<b>71.8%</b>	<b>63.0%</b>	<b>27.2%</b>
Max correlation between two aspects	<b>73.4%</b>	<b>86.5%</b>	<b>29.8%</b>

Table 2.6: Performance of the multi-aspect sentiment classification task for the *decorrelated Beer* dataset. MTM still outperforms baselines in the unrealistic decorrelated settings.

				F1 Score			
Interp.		Model	Params	Macro	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
None	SENT	Sentiment Majority	426k	68.89	67.48	73.49	65.69
	BASE	Emb <sub>200</sub> + Enc <sub>CNN</sub> + Clf	173k	78.23	78.38	80.86	75.47
Coarse-grained	SAA	Emb <sub>200</sub> + Enc <sub>CNN</sub> + A <sub>Shared</sub> + Clf	196k	78.19	77.43	80.96	76.16
		Emb <sub>200</sub> + Enc <sub>LSTM</sub> + A <sub>Shared</sub> + Clf	186k	78.16	75.88	81.25	77.36
Fine-grained	NB-SVM	(Wang and Manning, 2012)	3 · 426k	74.60	73.50	77.32	72.99
	SAM	(Lei et al., 2016)	3 · 644k	77.06	77.36	78.99	74.83
	MASA	Emb <sub>200</sub> + Enc <sub>LSTM</sub> + A <sub>Aspect-wise</sub> <sup>Sparse</sup> + Clf	458k	78.82	77.35	81.65	77.47
	MAA	Emb <sub>200</sub> + Enc <sub>LSTM</sub> + A <sub>Aspect-wise</sub> + Clf	458k	78.96	78.54	81.56	76.79
	MTM	Emb <sub>200</sub> + Masker + Enc <sub>CNN</sub> + Clf (Ours)	274k	79.32	78.58	81.71	77.66
	MTM <sup>C</sup>	Emb <sub>200+4</sub> + Enc <sub>CNN</sub> + Clf (Ours)	175k	79.66	78.74	82.02	78.22

the *Beer* reviews from Lei et al. (2016); Chang et al. (2020). As shown in Table 2.5, it has an average correlation of 27.2% and the aspect *Taste* is removed.

We find similar trends but stronger results (see Table 2.6): *MTM* significantly generates better rationales and achieves higher F1 scores than *SAM* and the attention models. The contextualized variant *MTM*<sup>C</sup> further improves the performance. Some visualizations are available in Appendix C.2.

### 2.6.3 Impact of the Length of Rationales

In this section, we study the distribution of the rationales’ lengths and their impact on the rationale performance in terms of the precision, recall, and F1 score. In Section 2.6.1, and for a fair comparison with prior work, we report only the precision on three aspects at a similar

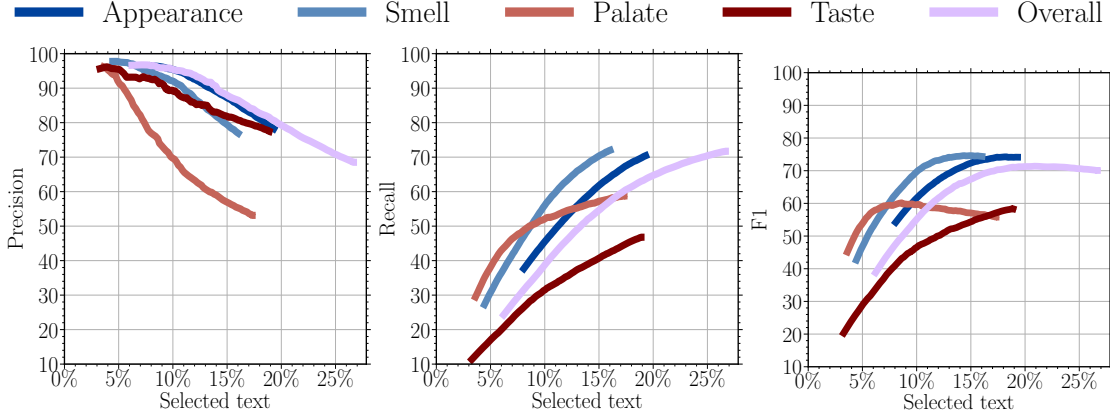


Figure 2.5: Performance related to human evaluation, showing the precision, recall, and F1 scores of the rationales for each aspect (including the overall rating) of the *Beer* dataset. The percentage of words indicates the number of highlighted words of the full review.

number of selected words on the human sentence-level aspect annotations.

We use the *Beer* dataset and also consider the extra aspect *Overall* (which leads to five aspects in total). We train *MTM* on it as in Section 2.5.1. We compute the distribution of the explanation length (in percent) on the validation set for each aspect  $a_i, i = 1, \dots, 5$ . We calculate all the 100 percentiles  $P_p^{a_i}$ . Then, we infer the sub-masks  $M_{a_i}$  and generate the rationale by selecting each word  $x^\ell$ , whose relevance towards  $a_i$  satisfies  $P(m_{a_i}^\ell | x^\ell) \geq P_p^{a_i}$ . Finally, we compute the precision, recall, and F1 score on the human annotation for each percentile  $P_p^{a_i}$ .

Figure 2.5 shows the result of the five aspects. First, we observe that the length distribution of the rationales for the aspect *Overall* is the most spread compared to the other aspects, and the one of the aspect *Smell* the least. The other three aspects share a similar distribution.

In terms of precision, we notice that the aspect *Palate* drops quickly compared to the other ones, that decrease linearly when we augment the portion of selected text. According to RateBeer<sup>6</sup>, the aspect *Palate* is the most difficult one to rate, which confirms our findings. For most aspects the precision remains high after selecting 75% of the text (5 - 15%), showing the effectiveness of our approach. In terms of recall, they all increase unsurprisingly when more words are highlighted.

Finally, we show that *MTM* generally achieves a very high precision ( $> 90\%$ ) by highlighting only 5 - 15% words of the input text, which reduces the cognitive load of users to identify the important parts of documents.

<sup>6</sup><https://www.ratebeer.com/Story.asp?StoryID=103>.

## 2.7 Conclusion

Providing explanations for automated predictions carries much more impact, increases transparency, and might even be necessary. Past work has proposed using attention mechanisms or rationale methods to explain the prediction of a target variable. The former produce noisy explanations, while the latter do not properly capture the multi-faceted nature of useful rationales. Because of the non-probabilistic assignment of words as justifications, rationale methods are prone to suffer from ambiguities and spurious correlations and thus, rely on unrealistic assumptions about the data.

The Multi-Target Masker (MTM) addresses these drawbacks by replacing the binary mask with a probabilistic multi-dimensional mask (one dimension per target), learned in an unsupervised and multi-task learning manner, while jointly predicting all the target variables. According to comparison with human annotations and automatic evaluation on two real-world datasets, the inferred masks were more accurate and coherent than those that were produced by the state-of-the-art methods. It is the first technique that delivers both the best explanations and highest accuracy for multiple targets simultaneously.

In Chapter 4, we use inferred masks for recommendation: to generate personalized and relevant explanations to support recommendations and build better user and item profiles. As future work, because masks embed opinions in reviews, they could be used to produce structured summaries across multiple aspects. Also, our method can be applied to different types of inputs like images. In that case, the words in the text data are replaced by regions or patches in an image, obtained by a classical pretrained convolutional neural network. The sparsity constraint stays similar, and the continuity is based on two-dimensional distances instead.





## 3 Rationalization through Concepts

### 3.1 Preface

**Contribution and Sources.** This chapter is largely based on Antognini and Faltings (2021b). The detailed individual contributions are listed below using the CRediT taxonomy (Brand et al., 2015) (terms are selected as applicable).

Diego Antognini (author): Conceptualization, Methodology, Software, Validation, Investigation, Formal Analysis, Writing – Original Draft, Writing – Review & Editing.

Boi Faltings: Writing – Review & Editing, Administration, Supervision.

**Summary.** Automated predictions require explanations to be interpretable by humans. One type of explanation is a rationale, i.e., a selection of input features such as relevant text snippets from which the model computes the outcome. However, a single overall selection does not provide a complete explanation, e.g., weighing several aspects for decisions. In Chapter 2, we use several aspect ratings to infer multi-dimensional rationales. However, in some cases, the availability of those labels is not guaranteed. To this end, we present in this chapter a novel self-interpretable model called ConRAT. Inspired by how human explanations for high-level decisions are often based on key concepts, ConRAT extracts a set of text snippets as concepts and infers which ones are described in the document. Then, it explains the one outcome with a linear aggregation of concepts. Two regularizers drive ConRAT to build interpretable concepts. In addition, we propose two techniques to boost the rationale and predictive performance further. Experiments on both single- and multi-aspect sentiment classification tasks show that ConRAT is the first to generate concepts that align with human rationalization while using only the overall label. Further, it outperforms state-of-the-art methods trained on each aspect label independently.

### 3.2 Introduction

Neural models have become the standard for many tasks, owing to their large performance gains. However, their adoption in decision-critical fields is more limited because of their lack of interpretability, particularly with textual data.

One of the simplest means of explaining predictions of complex models is by selecting relevant input features. Attention mechanisms (Bahdanau et al., 2015) model the selection using a conditional importance distribution over the inputs, but the resulting explanations are noisy (Jain and Wallace, 2019; Pruthi et al., 2020). Multi-head attention (Vaswani et al., 2017) extends attention mechanisms to attend information from different perspectives jointly. However, no explicit mechanisms guarantee a logical connection between different views (Voita et al., 2019; Kovaleva et al., 2019). Another line of research includes rationale generation methods (Lei et al., 2016; Chang et al., 2020; Antognini et al., 2021b). If the selected text input features are short and concise – called a rationale – and suffice on their own to yield the prediction, it can potentially be understood and verified against domain knowledge (Chang et al., 2019).

The key motivation for this chapter arises from the limitations of rationales. Rationalization models strive for one overall selection to explain the outcome by maximizing the mutual information between the rationale and the label. However, useful rationales can be multi-faceted, where each facet relates to a particular “concept” (see Figure 3.1). For example, users typically justify their opinions of a product by weighing explanations: one for each aspect they care about (Musat et al., 2013; Musat and Faltings, 2015).

Inspired by how human reasoning comprises concept-based thinking (Armstrong et al., 1983; Tenenbaum, 1999), we aim to discover, in an unsupervised manner, a set of concepts to explain the outcome with a weighted average, similar to multi-head attention. In this chapter, we relate concepts to semantically meaningful and consistent excerpts across multiple texts. Unlike topic modeling, where documents are described by a set of latent topics comprising word distributions, our latent concepts relate to text snippets that are relevant for the prediction.

Another motivation for this chapter is to generate interpretable concepts. The explanation of an outcome should rely on concepts that satisfy the desiderata introduced in Alvarez-Melis and Jaakkola (2018). They should

1. preserve relevant information,
2. not overlap with each other and be diverse, and
3. be human-understandable.

Figure 3.1 shows an example of concepts in the beer domain.

In this chapter, we present a novel self-explaining neural model: the concept-based rationalizer (ConRAT) (see Figure 3.1 and Figure 3.2). Our new rationalization scheme first identifies a set of concepts in a document and then decides which ones are currently described (bi-

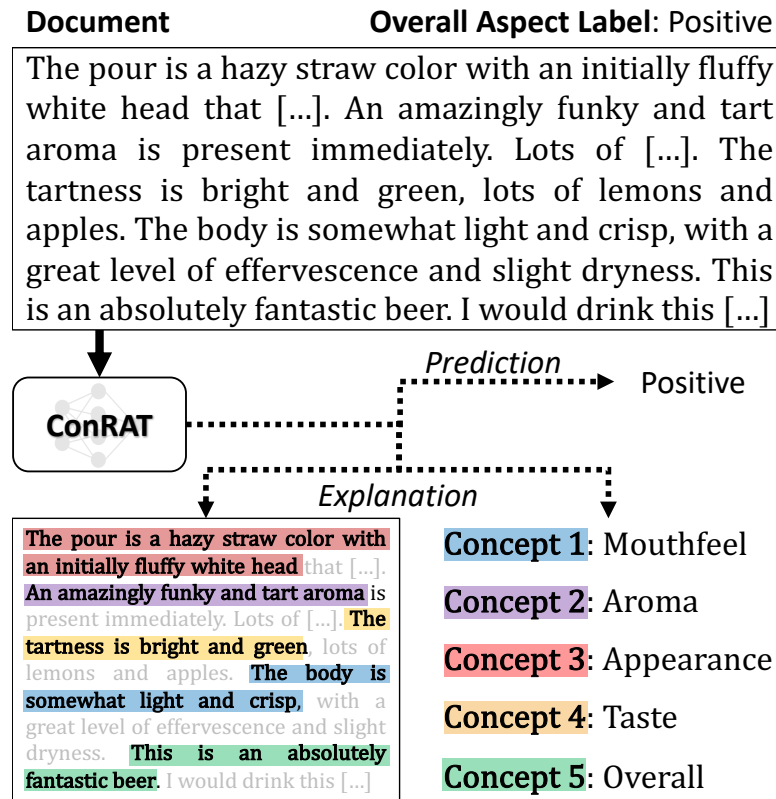


Figure 3.1: An illustration of ConRAT. Given a beer review, ConRAT identifies five excerpts that relate to particular concepts of beers (i.e., the explanation), depicted in color, from which it computes the outcome.

nary selection). ConRAT explains the prediction with a linear aggregation of concepts. The model is trained end-to-end, and the concepts are learned in an unsupervised manner. In addition, we design two regularizers that guide ConRAT to induce interpretable concepts and propose two optional techniques, knowledge distillation and concept pruning, in order to boost the performance further.

We evaluate ConRAT on both single- and multi-aspect sentiment classification with up to five target labels. Upon training ConRAT only on the overall aspect, the results show that ConRAT generates concepts that are relevant, diverse, and non-overlapping, and they also recover human-defined concepts. Furthermore, our model significantly outperforms strong supervised baseline models in terms of predictive and explanation performance.

### 3.3 Related Work

Developing interpretable models is of considerable interest to the broader research community. Researchers have investigated many approaches to improve the interpretability of neural networks.

### 3.3.1 Interpretability

The first line of research aims at providing post-hoc explanations of an already trained model. For example, gradient and perturbation-based methods attribute the decision to important input features (Ribeiro et al., 2016; Sundararajan et al., 2017; Lundberg and Lee, 2017; Shrikumar et al., 2017). Other studies identified the causal relationships between input-output pairs (Alvarez-Melis and Jaakkola, 2017; Goyal et al., 2019). In contrast, our model is inherently interpretable as it directly produces the prediction with an explanation.

Another line of research has developed interpretable models. Quint et al. (2018) extended a variational autoencoder with a differentiable decision tree. Alaniz and Akata (2019) proposed an explainable observer-classifier framework whose predictions can be exposed as a binary tree. However, these methods have been designed for images only, while our work focuses on text input.

The works most relevant to ours relate to interpretable models from the rationalization field (Lei et al., 2016; Bastings et al., 2019; Yu et al., 2019; Chang et al., 2020; Jain et al., 2020; Paranjape et al., 2020). These methods justify their predictions by selecting rationales (i.e., relevant tokens in the input text). However, they are limited to explain only the prediction with mostly one text span and rely on the assumption that the data have low internal correlations (Antognini et al., 2021b). Chang et al. (2019) extended previous methods to extract an additional rationale in order to counter the prediction. In our work, ConRAT produces multifaceted rationales and explains the prediction through a linear aggregation of the extracted concepts. However, if we set the number of concepts to one, ConRAT reduces to a special case of a rationale model.

### 3.3.2 Explanations through Concepts

Researchers have proposed multiple approaches for concept-based explanations. Kim et al. (2018) designed a post-hoc technique to learn concept activation vectors by relying on human annotations that characterize concepts of interest. Similarly, Bau et al. (2017); Zhou et al. (2018) generated visual explanations for a classifier. Our concepts are learned in an unsupervised manner and not defined a priori.

Few studies have learned concepts on images in an unsupervised fashion. Li et al. (2018b) explained predictions based on the similarity of the input to “prototypes” learned during training. Alvarez-Melis and Jaakkola (2018) used an autoencoder to extract relevant concepts and explain the prediction. Ghorbani et al. (2019) designed an unsupervised concept discovery method to explain trained models. Koh et al. (2020) employed the discovered concepts to predict the target label. Our work’s key difference is that we focus on text data, while all these methods treat only image inputs.

To the best of our knowledge, Bouchacourt and Denoyer (2019) is the only study that has proposed a self-interpretable concept-based model for text data using reinforcement learning.

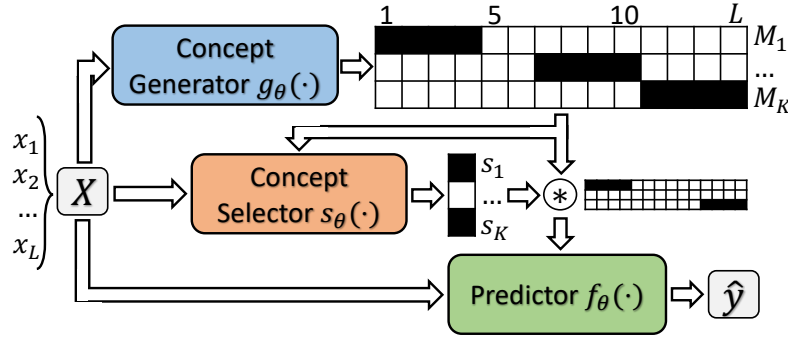


Figure 3.2: The proposed self-explaining model ConRAT. The model predicts and explains  $\hat{y}$ . Given a document  $X$ , the concept generator produces one binary mask per concept. The concept selector decides which concepts are present in the input. The predictor aggregates each selected concept’s prediction to compute  $\hat{y}$ .

It computes the predictions and provides an explanation in terms of the presence or absence of concepts in the input (i.e., text excerpts of variable lengths). However, their method achieves poor overall performance. In addition, it is unclear whether the discovered concepts are interpretable. Conversely, ConRAT is differentiable, clearly outperforms strong models in terms of predictive and explanation performance, and it infers relevant, diverse, non-overlapping, and human-understandable concepts.

### 3.3.3 Topic Modeling

Topic models, such as latent Dirichlet allocation (Blei et al., 2003), describe documents with a mixture of latent topics. Each topic represents a word distribution. Some studies combined topic models with recurrent neural models (Dieng et al., 2017; Zaheer et al., 2017). However, the goal of these generative models and the topics remains different than this chapter’s. We aim to build a self-interpretable model that predicts and explains the outcome with latent concepts.

## 3.4 ConRAT: A Concept-based Rationalizer

Figure 3.2 depicts the architecture of our proposed self-explaining model: the Concept-based Rationalizer (ConRAT). Let  $X$  be a random variable representing a document composed of  $T$  words  $(x_1, x_2, \dots, x_T)$ ,  $y$  the ground-truth label, and  $K$  the desired numbers of concepts<sup>1</sup>. Given a document  $X$  and a label  $y$ , our goal is to explain the prediction  $\hat{y}$  by finding a set of  $K$  concepts  $C_1, \dots, C_K$  that are masked versions of  $X$ . ConRAT learns concepts by maximizing the mutual information between  $\mathbf{C}$  and  $y$ . We guide ConRAT to create separable and consistent concepts via two regularizers to make them human-understandable.

<sup>1</sup>Our method is easily adapted for regression problems.

### 3.4.1 Model Overview

ConRAT is divided into three submodels: a **Concept Generator**  $g_\theta(\cdot)$ , which finds the concepts  $C_1, \dots, C_K$ ; a **Concept Selector**  $s_\theta(\cdot)$ , which detects whether a concept  $C_k$  is present or absent (i.e.,  $s_k \in \{1, 0\}$ ) in the input  $X$ ; and a **Predictor**  $f_\theta(\cdot)$ , which predicts the outcome  $\hat{y}$  based on the concepts  $\mathbf{C}$  and their presence scores  $\mathbf{S}$ .

#### Concept Generation.

Inspired by the selective rationalization field (Lei et al., 2016), we define “concept” as a sequence of consecutive words in the input text. Previous studies extracted only one concept  $C_1$  that is sufficient to explain the target variable  $y$ . In our work, a major difference is that we aim to find  $K$  concepts  $C_1, \dots, C_K$  that represent different topics or aspects and altogether explain the target variable  $y$ . We interpret the model as being linear in the concepts rather than depending on one overall selection of word. More formally, we define a concept as follows:

$$C_k = M_k \odot X, \quad (3.1)$$

where  $M_k \in \mathbb{S}$  denotes a binary mask,  $\mathbb{S}$  is a subset of  $\mathbb{Z}_2^T$  with some constraints (introduced in Section 3.4.2), and  $\odot$  is the element-wise multiplication of two vectors.

We parametrize the binary masks  $\mathbf{M} \in \mathbb{Z}_2^{K \times T}$  with the concept generator model  $g_\theta(\cdot)$ , based on a bi-directional recurrent neural network. Following previous rationalization research (Yu et al., 2019; Chang et al., 2020), we force  $g_\theta(\cdot)$  to select one chunk of text per concept with a pre-specified length  $\ell \in [1, T]^2$ . Instead of predicting the mask  $M_k$  directly,  $g_\theta(\cdot)$  produces a score for each position  $t$ . Then, it samples the start position  $t_k^*$  of the chunk for each  $C_k$  using the straight-through Gumbel-Softmax (Maddison et al., 2017; Jang et al., 2017). Finally, we compute  $M_k$  as follows:

$$\begin{aligned} T^* &\sim \text{Gumbel}(g_\theta(X)), \\ M_{k,t} &= \mathbb{1}[t \in [t_k^*, \min(t_k^* + \ell - 1, T)]], \end{aligned} \quad (3.2)$$

where  $\mathbb{1}$  denotes the indicator function. Although the equation is not differentiable, we can employ the straight-through technique (Bengio et al., 2013) and approximate it with the gradient of a causal convolution and a convolution kernel of an all-one vector of length  $\ell$ .

#### Concept Selection.

A key objective of ConRAT is to produce semantically consistent and separable concepts. So far, the generator  $g_\theta(\cdot)$  generates  $K$  concepts for any input document. However, some documents

---

<sup>2</sup>In early experiments, we relaxed the length constraint and generated instead  $K$  differentiable masks with continuity regularizers. However, this variant produced majorly inferior results. We hypothesize that there are too many constraints to optimize with only the target label as a strong signal.

might mention only a subset of those. Thus, the goal of the concept selector model  $s_\theta(\cdot)$  is to enable ConRAT to ignore absent concepts.

Specifically, for each concept  $C_k$ , the model first computes a concept representation  $H_{C_k}$  using a standard attention mechanism (Bahdanau et al., 2015) (the tokens whose  $M_{k,t} = 0$  are masked out). Then, we take the dot product of  $H_{C_k}$  with a weight vector, followed by a sigmoid activation function to induce the log-probabilities of a relaxed Bernoulli distribution (Jang et al., 2017). Finally, we sample the presence score  $s_k \in \{0, 1\}$  of each concept independently:

$$S \sim \text{RelaxedBernoulli}(s_\theta(X, \mathbf{M})), \quad (3.3)$$

where  $\mathbf{M}$  denote the binary masks and  $X$  the input tokens.

### Prediction.

As inputs, the predictor  $f_\theta(\cdot)$  takes the document  $X$ , the masks  $\mathbf{M}$ , and the presence scores  $S$  for all concepts. First, we extract the concepts, which are masked versions of  $X$ . Differently than in Equation 3.1, the concepts are ignored if  $s_k = 0$ :

$$C_k = (M_k * s_k) \odot X. \quad (3.4)$$

Second, the model produces the hidden representation  $h'_{C_k}$  with another recurrent neural network, followed by a LeakyReLU activation function (Xu et al., 2015a). Then, it computes the logits of  $y$  by applying a linear projection for each concept:

$$P_k = W h'_{C_k} + b, \quad (3.5)$$

where  $W$  and  $b$  are the projection parameters. Finally,  $f_\theta(\cdot)$  computes the final outcome as follows:

$$p(y|C, \mathbf{M}, X) = \text{softmax}\left(\sum_{k=1}^K \alpha_k P_k s_k\right), \quad (3.6)$$

where  $\alpha_k$  are model parameters that can be interpreted as the degree to which a particular concept contributes to the final prediction.

### 3.4.2 Unsupervised Discovery of Concepts

The above formulations integrate the explanation into the outcome computation. However,  $M_k$  is by definition faithful to the model's inner workings but not comprehensible for the end user (Jacovi and Goldberg, 2020). Following Alvarez-Melis and Jaakkola (2018), we aim the concepts to follow three desiderata:

1. **Fidelity**: they should preserve relevant information,
2. **Diversity**: they should be non-overlapping and diverse, and

3. **Grounding:** they should have an immediate human-understandable interpretations.

The hard constraint in Equation 3.2 naturally enforces the grounding by forcing the concept to be a sequence of  $\ell$  words. For the fidelity, it is partly integrated in ConRAT by the prediction loss, which is the cross-entropy between the ground-truth label  $y$  and the prediction  $\hat{y}$ :  $\mathcal{L}_{pred} = CE(\hat{y}, y)$ . Recall that the concepts are substitutes of the input that are sufficient for the prediction. We emphasize the word “partly” because nothing prevents ConRAT from picking up spurious correlations.

We propose two regularizers to encourage ConRAT in finding non-overlapping, relevant, and dissimilar concepts. The first favors the orthogonality of concepts by penalizing redundant rows in  $\mathbf{M}$ :

$$\mathcal{L}_{overlap} = \|\mathbf{M}\mathbf{M}^T - \ell \cdot \mathbb{1}\|_F^2, \quad (3.7)$$

where  $\|\cdot\|_F$  stands for the Frobenius norm of a matrix,  $\mathbb{1}$  denotes the identity matrix, and  $\ell$  the pre-specified concept length. However,  $\mathcal{L}_{overlap}$  alone does not prevent ConRAT from learning little relevant concepts. Therefore, we propose a second regularizer to encourage fidelity and diversity by minimizing the cosine similarity between the concept representations  $H_{C_k}$  (see Section 3.4.1):

$$\mathcal{L}_{div} = \frac{1}{K} \frac{1}{K-1} \sum_{\substack{k_1, k_2=1 \\ k_1 \neq k_2}}^K \cos(H_{C_{k_1}}, H_{C_{k_2}}). \quad (3.8)$$

In both regularizers, we do not consider the presence scores  $S$  because a model could always select only one concept; this strategy is suboptimal and reduces to a special case of rationale models (i.e.,  $S$  would become a one-hot vector).

To summarize, the concepts are learned in an unsupervised manner and align with the three desiderata mentioned above: diversity is achieved with  $\mathcal{L}_{overlap}$  and  $\mathcal{L}_{div}$ ; fidelity is enforced by  $\mathcal{L}_{pred}$  and  $\mathcal{L}_{div}$ , and the hard constraint in Equation 3.2 ensures the grounding. Finally, we train ConRAT end-to-end and minimize the loss jointly  $\mathbf{L} = \mathcal{L}_{pred} + \lambda_O \mathcal{L}_{overlap} + \lambda_D \mathcal{L}_{div}$ , where  $\lambda_O$  and  $\lambda_D$  control the impact of each regularizer.

### 3.4.3 Improving Overall Performance Further

The purpose of self-explaining models is to compute outcomes while being more interpretable. However, one key point is to achieve predictive performance comparable to that of black-box models. We propose two techniques to further improve both interpretability and performance; however, ConRAT does not require these techniques to outperform other methods, as we will see later.

**Knowledge Distillation.** We can train ConRAT not only via the information provided by the true labels but also by observing how a teacher model behaves (Hinton et al., 2015). In that



case, we introduce the teacher model  $T_\theta(\cdot)$ , which is a simple recurrent neural network similar to the predictor  $f_\theta(\cdot)$ . It is trained on the same data, but it uses the whole input  $X$  instead of subsets selected by each  $C_k$ . The overall training loss becomes  $\mathbf{L} = \mathcal{L}_{pred} + \lambda_O \mathcal{L}_{overlap} + \lambda_D \mathcal{L}_{div} + \lambda_T (\hat{y}_{T_\theta} - \hat{y}_{f_\theta})^2$ .

**Pruning Concepts.** Depending on the number of concepts and the pre-specified length, the total number of selected words can be close to or higher than the document length<sup>3</sup>. In practice, it is hard to extract meaningful concepts in such settings. To alleviate this problem, we propose to prune concepts at inference and select the top-k concepts that overlap the least with the others. More specifically, we compute the overlap as follows: for each sample in the validation set, we measure the average overlap ratio between  $M_{k_1}$  and  $M_{k_2}$  for each concept-pair  $(C_{k_1}, C_{k_2})$ ,  $k_1 \neq k_2$ . Then, we select the top-k concepts whose scores are the lowest. Finally, to compute the new prediction  $\hat{y}$ , we update  $s_k = 1$  if  $C_k$  is in the top-k or  $s_k = 0$  otherwise.

## 3.5 Experiments

### 3.5.1 Datasets

We evaluate the quantitative performance of ConRAT using two binary classification datasets. The first one is the single-aspect Amazon Electronics dataset (Ni et al., 2019). We followed the filtering process in Chang et al. (2019) to keep only the reviews that contain evidence for both positive and negative sentiments. Specifically, we considered the first 50 tokens after the words “pros:” and “cons:” as the rationale annotations for the positive and negative labels, respectively. We randomly picked 24,000 balanced samples with ratings of four and above or two and below.

The second dataset comprises the multi-aspect beer reviews (McAuley et al., 2012) used in the field of rationalization (Lei et al., 2016; Yu et al., 2019). Each review describes various beer aspects: Appearance, Aroma, Palate, Taste, and Overall; users also provided a five-star rating for each aspect. However, we only use the overall rating for ConRAT. The dataset includes 994 beer reviews with sentence-level aspect annotations. Following the evaluation protocol in Bao et al. (2018); Chang et al. (2020), we binarized the ratings  $\leq 2$  as negative and  $\geq 3$  as positive. We sampled 60,000 balanced examples. Our setting is more challenging than those in previous studies because we assess the performance on all aspects (instead of three) and consider all examples for the sampling (instead of de-correlated subsets), reflecting the real data distribution. Table 3.1 shows the data statistics.

<sup>3</sup>e.g., if a document contains 200 tokens and we aim to extract 10 concepts of 20 tokens, all words should be selected.

Table 3.1: Statistics of the review datasets.

Dataset	Amazon	Beer
# Reviews	24,000	60,000
Split Train/Val/Test	20k/2k/2k	50k/5k/5k
# Annotations	471	994
# Human Aspects	1	5
# Words per review	$224 \pm 125$	$184 \pm 58$
# Sentences per review	$12.0 \pm 7.6$	$12.6 \pm 4.9$

### 3.5.2 Baselines

We consider the following baselines. **RNP** is a generator-predictor framework proposed by Lei et al. (2016) for rationalizing neural prediction. The generator selects text spans as rationales, which are then fed to the classifier for the final prediction. Yu et al. (2019) introduced **RNP-3P**, which extends RNP to include the complement predictor as the third player. It maximizes the predictive accuracy from unselected words. The training consists of an adversarial game with the three players. **Intro-3P** (Yu et al., 2019) improves RNP-3P by conditioning the generator on the predicted outcome of a teacher model. **InvRAT** is a game-theoretic method that competitively rules out spurious words with strong correlations to the output. The game-theoretic approach **CAR** aims to infer a rationale and a counterfactual rationale that counters the true label. We follow Chang et al. (2020) and consider for all methods their hard constraint variant (i.e., selecting one chunk of text) with different lengths for generating rationales.

RNP-3P and Intro-3P are trained with the policy gradient (Williams, 1992). The others estimate the gradients of the rationale selections using the straight-through technique (Bengio et al., 2013).

All rationalization methods, except CAR, strive for a single overall selection ( $K = 1$ ) to explain the outcome. For the multi-aspect dataset, we train and tune each baseline independently for each aspect. The key difference with ConRAT is that the model is only trained on the overall aspect label and infers one rationale of  $K$  concepts; the baselines are trained  $K$  times to infer one rationale of one concept.

### 3.5.3 Experimental Details

To seek fair comparisons, we try to keep a similar number of parameters across all models, and we employ the same architecture for each player (generators, predictors, and discriminators/teachers) in all models: bi-directional gated recurrent units (Chung et al., 2014) with a hidden dimension 256. We use the 100-dimensional GloVe word embeddings (Pennington et al., 2014), Adam (Kingma and Ba, 2015) as optimization method with a learning rate of 0.001. We set the convolutional neural network in the concept selector similarly to Kim et al. (2015) with 3-, 5-, and 7-width filters and 50 feature maps per filter. For ConRAT, we set the regularizer

Table 3.2: Accuracy and objective performance of rationales in automatic evaluation for the Amazon dataset. ConRAT infers rationales that align with factual and counter factual ones.

Model	Acc.	Factual			Counter Factual		
		P	R	F	P	R	F
RNP	<b>75.5</b>	32.6	18.8	23.8		–	
RNP-3P	70.0	49.4	28.4	36.0		–	
Intro-3P	75.2	22.1	12.8	16.2		–	
InvRAT	71.5	44.3	25.5	32.4		–	
ConRAT-1 (Ours)	<b>75.5</b>	<b>56.4</b>	<b>32.5</b>	<b>41.3</b>		–	
CAR	73.6	33.0	19.1	24.2	<b>44.1</b>	<b>25.4</b>	<b>32.2</b>
ConRAT-6 (Ours)	<b>75.4</b>	<b>50.0</b>	<b>28.8</b>	<b>36.6</b>	32.3	18.6	23.6
ConRAT-4 (Ours)	75.3	46.4	26.7	33.9	29.6	17.1	21.6
ConRAT-2 (Ours)	75.3	33.7	19.4	24.6	8.9	5.1	6.5

factors as follow:  $\lambda_O = 0.05$ ,  $\lambda_D = 0.05$ , and  $\lambda_T = 0.5$ . We use the open-source implementation for all models, and we tune them by maximizing the prediction accuracy on the dev set with 16 random searches. For reproducibility purposes, we include additional details in Appendix F.2.

### 3.5.4 RQ 1: Can ConRAT find evidence for factual and counterfactual rationales?

We aim to validate whether ConRAT can identify the two evidences for positive and negative sentiments. We set the concept length  $\ell = 30$ , we compare the generated rationales with the annotations, and we report the precision, recall, and F1 score. In this experiment, no teacher is used in ConRAT.

Table 3.2 contains the results. The top rows contain the results when only the factual rationales are considered for the evaluation, and ConRAT-1 uses only one concept. We see that ConRAT surpasses the baselines in finding rationales that align with human annotations, and it also matches the test accuracy with the baselines. Interestingly, we note that the baselines achieving the highest accuracy underperform in finding the correct rationales.

For the factual and counterfactual rationales, CAR finds one rationale to support the outcome and another one to counter it, in an adversarial game. However, the concepts inferred by ConRAT are not guaranteed to align with the rationales as there is no explicit signal to infer counterfactual concepts. Thus, we increase the number of concepts up to six and prune ConRAT to consider only the two most dissimilar concepts (see Section 3.4.3).

The bottom of Table 3.2 shows the results. With only two concepts, ConRAT-2 outperforms CAR in terms of test accuracy and matches the performance for the factual rationales, but it poorly identifies counterfactual rationales. However, there is a major improvement when we increase the number of concepts and use pruning. Indeed, the word distribution of the factual and counterfactual rationales are different, hence captured with pruning. ConRAT’s

### Chapter 3. Rationalization through Concepts

Table 3.3: Objective performance of rationales for the multi-aspect beer reviews. ConRAT only uses the overall label and ignores the other aspect labels. All baselines are trained separately on each aspect rating. **Bold** and underline denote the best and second-best results, respectively. ConRAT is competitive with the fully-supervised baselines.

Model	Acc.	Average			Appearance			Aroma			Palate			Taste			Overall			
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	
$\ell = 20$	RNP	81.1	30.7	22.1	24.9	30.8	23.2	26.5	22.1	21.0	21.5	17.7	24.1	20.4	28.1	16.7	20.9	54.9	25.8	35.1
	RNP-3P	80.5	29.1	22.5	25.0	30.4	25.6	27.8	19.3	20.4	19.8	10.3	12.0	11.1	43.9	<u>28.4</u>	<u>34.5</u>	41.6	26.0	32.0
	Intro-3P	<u>85.6</u>	24.2	19.6	21.3	28.7	24.8	26.6	14.3	14.4	14.3	16.6	19.3	17.9	24.2	13.6	17.4	37.0	25.9	30.5
	InvRAT	82.9	<u>41.8</u>	<u>31.1</u>	<u>34.8</u>	<u>54.5</u>	<u>45.5</u>	<u>49.6</u>	<u>26.1</u>	<u>27.6</u>	<u>26.9</u>	<u>22.6</u>	<u>25.9</u>	<u>24.1</u>	<u>46.6</u>	27.4	34.5	<u>59.0</u>	<u>29.3</u>	<u>39.2</u>
	ConRAT*(Ours)	<b>91.4</b>	<b>50.0</b>	<b>42.0</b>	<b>44.9</b>	<b>57.8</b>	<b>53.0</b>	<b>55.3</b>	<b>31.9</b>	<b>35.5</b>	<b>33.6</b>	<b>29.0</b>	<b>36.3</b>	<b>32.3</b>	<b>56.5</b>	<b>33.9</b>	<b>42.4</b>	<b>74.9</b>	<b>51.0</b>	<b>60.7</b>
$\ell = 10$	RNP	<u>84.4</u>	41.3	16.6	23.2	40.1	12.0	18.5	<b>33.3</b>	<b>18.7</b>	<b>24.0</b>	<b>25.1</b>	<b>17.4</b>	<b>20.6</b>	32.3	9.8	15.07	76.0	25.1	37.8
	RNP-3P	83.1	31.1	13.5	18.6	41.8	19.2	26.3	22.2	12.4	15.9	16.5	10.4	12.7	33.2	10.6	16.1	41.9	14.7	21.8
	Intro-3P	80.9	21.8	10.8	14.3	51.0	26.0	34.4	18.8	9.7	12.8	16.5	10.6	12.9	9.7	2.6	4.1	13.1	5.2	7.4
	InvRAT	81.9	<u>47.1</u>	<u>17.8</u>	<u>25.5</u>	<b>59.4</b>	<u>26.1</u>	<b>36.3</b>	31.3	15.5	20.8	16.4	9.6	12.1	<u>39.1</u>	<u>11.6</u>	<u>17.9</u>	<b>89.1</b>	<u>26.4</u>	<u>40.7</u>
	ConRAT*(Ours)	<b>91.3</b>	<b>48.1</b>	<b>20.1</b>	<b>28.0</b>	<u>51.7</u>	<b>26.2</b>	<u>34.8</u>	<u>32.6</u>	<u>17.4</u>	<u>22.7</u>	<u>23.0</u>	<u>13.8</u>	<u>17.3</u>	<b>45.3</b>	<b>13.1</b>	<b>20.3</b>	<u>88.0</u>	<b>30.1</b>	<b>44.9</b>

\* The model is only trained on the overall label and does not have access to the other ground-truth labels.

factual rationales are better than those of all models. The counterfactual ones get closer to those produced by CAR. We show later in Section 3.5.6 that pruning helps in achieving better correlation with human judgments but is not required.

#### 3.5.5 RQ 2: Are concepts inferred by ConRAT consistent with human rationalization?

We investigate whether ConRAT can recover all beer aspects by using only the overall ratings. Because beer reviews are smaller in length than Amazon ones, we set the concept length  $\ell$  to 10 and 20. We fix the number of concepts to ten and prune ConRAT to keep five. We manually map them to the closest aspect for comparison. We trained the teacher model, used in Intro-3P and ConRAT, and obtained 91.4% accuracy. More results and illustrations are available in Appendix A.1 and Appendix C.3, respectively.

#### Objective Evaluation.

Similar to Section 3.5.4, we compare the generated rationales with the human annotations on the five aspects and the average performance. The main results are shown in Table 3.3. On average, ConRAT achieves the best performance while trained only on the overall ratings. This shows that the generated concepts, learned in an unsupervised manner, are separable, consistent, and correlated with human judgments to a certain extent. For the concept length  $\ell = 20$ , ConRAT produces significant superior results for all aspects, whereas the difference with InvRAT is less pronounced for  $\ell = 10$ . Finally, ConRAT’s concepts lead to the highest

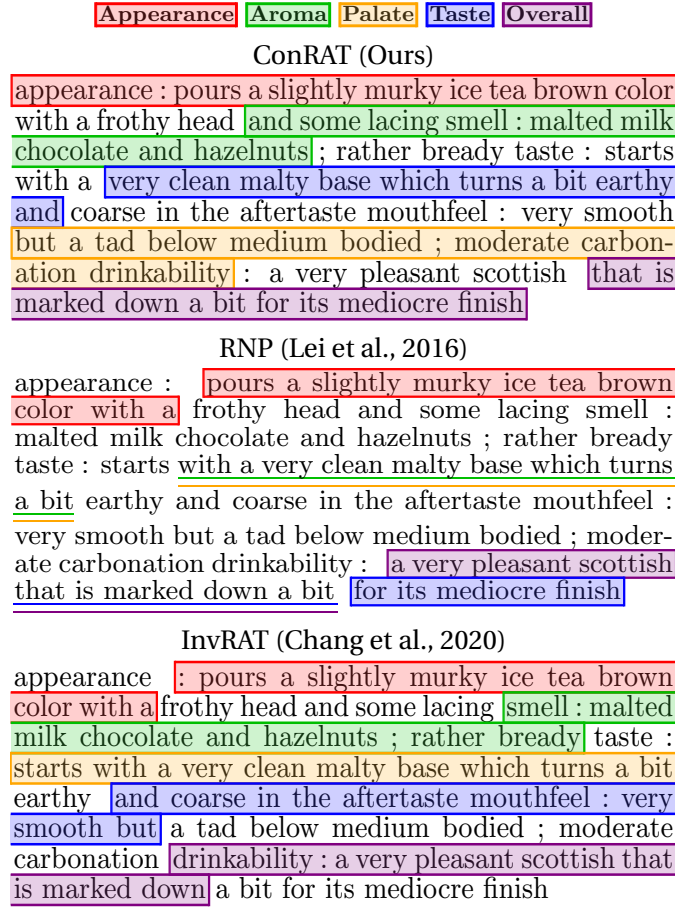


Figure 3.3: Concepts generated (with  $\ell=10$ ) for a beer review. Underline highlights ambiguities. The color depicts the aspects: Appearance, Aroma, Palate, Taste, and Overall. **ConRAT is trained only on the overall label.** RNP produces the same rationale for the aspect Aroma and Palate. InvRAT infers wrong rationales for the aspects Palate and Taste. In contrast, ConRAT reduces the impact of spurious correlations.

accuracy and respect the grounding desideratum, thanks to the teacher.

We hypothesize that the baselines underperform due to the high correlations among the aspect ratings. Thus, they are more prone to pick up spurious correlations between the input features and the output. By considering multiple concepts simultaneously, ConRAT reduces the impact of spurious correlations. Regarding Intro-3P and RNP-3P, both suffer from instability issues due to the policy gradient (Chang et al., 2020; Yu et al., 2019).

We visualize an example in Figure 3.3. We observe that ConRAT induces interpretable concepts, while the best baselines suffer from spurious correlations. By reading our concepts alone, humans will easily predict the aspect label and its polarity.

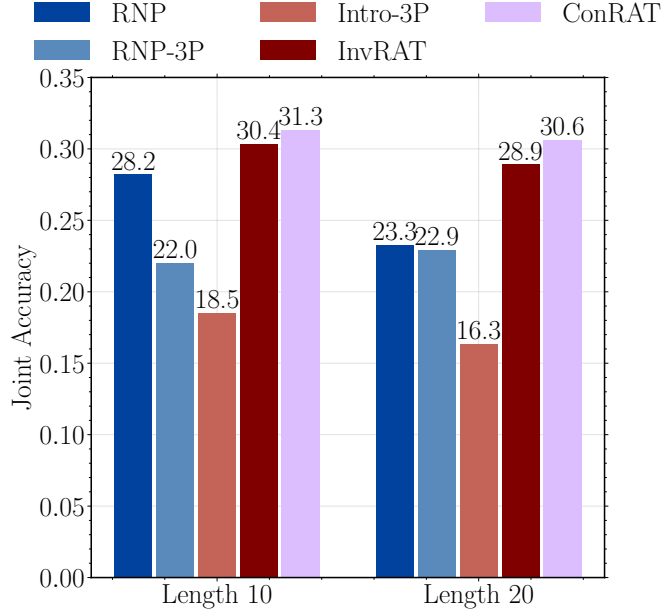


Figure 3.4: Subjective performance of rationales for the multi-aspect beer reviews. Evaluators need to guess both the sentiment and what aspect the concept is about, which makes random guess only 10%. ConRAT infers meaningful rationales according to human annotators.

#### Subjective Evaluation.

We conduct a human evaluation using Amazon’s Mechanical Turk (details in Appendix D.1) to judge the understandability of the concepts. Following Chang et al. (2019), we sampled 100 balanced reviews from the hold-out set for each aspect, model, and concept length, resulting in 5,000 samples. We showed the examples in random order. An evaluator is presented with the concept generated by one of the five methods (unselected words are not visible). We credit a success when the evaluator guesses the true aspect label and its sentiment. We report the success rate as the performance metric. A random guess has a 10% success rate.

Figure 3.4 shows the main results. Similar to the objective evaluation, ConRAT reaches the best performance, followed by InvRAT. Moreover, ConRAT only requires a single training on the overall aspect. It emphasizes that the discovered concepts satisfy the fidelity and diversity desiderata and better correlate with human judgments compared with supervised baselines. We study the error rates on each aspect. The Aroma and Palate aspects cause the highest for all models. One possible reason is that users confuse these with the aspect Taste, hence their high correlations in rating scores (Antognini et al., 2021b).

Table 3.4: Impact of the number of concepts in ConRAT on the objective performance for the beer reviews.

		<i>Average</i>			
#Concepts	Acc.	P	R	F	
$\ell = 20$	$K = 5$	90.95	48.96	37.59	41.37
	$K = 10$	<b>91.35</b>	<b>50.02</b>	<b>41.96</b>	<b>44.86</b>
	$K = 20$	90.24	37.78	31.19	32.84
$\ell = 10$	$K = 5$	89.64	47.60	19.23	26.90
	$K = 10$	<b>91.25</b>	<b>48.12</b>	<b>20.11</b>	<b>27.97</b>
	$K = 20$	91.05	35.71	14.84	20.71

Table 3.5: Ablation study of ConRAT with five concepts. The results confirm our intuition and that our regularizers enforce the interpretable desiderata.

Model	Acc.	Average		
		P	R	F
ConRAT	89.64	47.60	19.23	26.90
- No $\mathcal{L}_{overlap}$	91.05	31.50	13.16	18.37
- No $\mathcal{L}_{div}$	90.85	34.49	11.69	16.95
- No $s_{\theta}(\cdot) : s_k = 1 \forall k$	89.74	43.13	14.95	21.42
- No Teacher	86.52	45.31	19.65	26.99

### 3.5.6 RQ 3: How does the number of concepts $K$ in ConRAT affect the performance?

We study the impact of the number of concepts  $K$  in ConRAT on the performance, as discussed in Section 3.5.5. We set the number of concepts to the number of aspects ( $K=5$ ) and then increase it to  $K=10$  and  $K=20$ . We prune ConRAT to keep only the five most dissimilar concepts (see Section 3.4.3).

Results are shown in Table 3.4. First, we observe that the performance is already better than the baselines in Table 3.3 with  $K=5$ . Second, when increasing  $K$  and pruning ConRAT, the performance is boosted further. However, we remark that the interpretability of the concepts follows a bell curve and significantly decreases when  $K=20$ . One potential reason is that we expect overlaps between the discriminative concepts that relate to beer aspects<sup>4</sup>. Thus, the five most dissimilar concepts might align less with human-defined concepts.

<sup>4</sup>As shown in Table 3.1, the mean length of beer reviews is 184 words. With  $\ell=20$  and  $C=20$ , 400 words are highlighted.

### 3.5.7 RQ 4: How does each module of ConRAT contribute to the overall performance?

Finally, we analyze the importance of each module in an ablation study. To avoid any bias from pruning, we set the number of concepts to five<sup>5</sup>.

Table 3.5 shows the results. When ConRAT ignores the overlapping or the diversity regularizer, we observe a large drop in the rationale performance. This is expected as the diversity desideratum is not encouraged anymore. However, we remark that the sentiment prediction accuracy increases, which is possibly caused by spurious correlation with the ground-truth label. When all concepts are considered ( $s_k = 1 \forall k$ ), we note that the sentiment accuracy stays similar. However, the objective performance decreases by 10% for the precision and more than 20% for the recall and F1 score. These results align with prior work: users write opinions about the topics they care about (Musat and Faltings, 2015; Antognini et al., 2021a). ConRAT reduces the noise at training by selecting concepts described in the current document. Finally, the teacher model helps ConRAT to boost the sentiment accuracy by more than 3% absolute score, without affecting the rationale quality.

## 3.6 Conclusion

Providing explanations for automated predictions carries much more impact, increases transparency, and might even be vital. Previous works have proposed using rationale methods to explain the prediction of a target variable. However, they do not properly capture the multi-faceted nature of useful rationales. In this chapter, we proposed ConRAT, a novel self-explaining model that extracts a set of concepts and explains the outcome with a linear aggregation of concepts, similar to how humans reason. Our second contribution is two novel regularizers that guide ConRAT to generate interpretable concepts. Experiments on both single- and multi-aspect sentiment classification datasets show that ConRAT, by using only the overall label, is the first to provide superior rationale and predictive performance compared with supervised state-of-the-art methods trained for each aspect label. Moreover, ConRAT produces concepts considered superior in interpretability when evaluated by humans.

---

<sup>5</sup>We obtain similar results with  $K=10$  and  $K=20$ .



# **Iterative Critiquing of Explanations Part II**



## 4 Generating Explanations and Interacting with Them through Critiquing

### 4.1 Preface

**Contribution and Sources.** This chapter is largely based on Antognini et al. (2021a) and Petrescu et al. (2021). The detailed individual contributions are listed below using the CRediT taxonomy (Brand et al., 2015) (terms are selected as applicable).

Authors' contributions in Antognini et al. (2021a):

Diego Antognini (author): Conceptualization, Methodology, Software, Validation, Investigation, Formal Analysis, Writing – Original Draft, Writing – Review & Editing.

Claudiu Musat: Writing – Review & Editing (supporting), Supervision (supporting).

Boi Faltings: Writing – Review & Editing, Administration, Supervision.

Authors' contributions in Petrescu et al. (2021) (\* denotes equal contributions):

Diego Antognini\* (author): Conceptualization (lead), Methodology (lead), Software (equal), Writing – Review & Editing (lead).

Diana Petrescu\*: Software (equal), Writing – Original Draft, Writing – Review & Editing (supporting).

Boi Faltings: Conceptualization (supporting), Methodology (supporting), Administration, Supervision.

**Summary.** Using personalized explanations to support recommendations has been shown to increase trust and perceived quality. However, to actually obtain better recommendations, there needs to be a means for users to modify the recommendation criteria by interacting with the explanation. In this chapter, we present a novel explanation technique using aspect

markers that learns to generate personalized explanations of recommendations from review texts, and we show that human users significantly prefer these explanations over those produced by state-of-the-art techniques. Our work's most important innovation is that it allows users to react to a recommendation by critiquing the textual explanation: removing (symmetrically adding) certain aspects they dislike or that are no longer relevant (symmetrically that are of interest). The system updates its user model and the resulting recommendations according to the critique. This is based on a novel unsupervised critiquing method for single- and multi-step critiquing with textual explanations. Empirical results show that our system achieves good performance in adapting to the preferences expressed in multi-step critiquing and generates consistent explanations. Finally, we introduce four web interfaces helping users make decisions and find their ideal item. Because our system is model-agnostic (for both recommender and critiquing models), it allows great flexibility and further extensions. Our interfaces are, above all, a useful tool for research on recommendation with critiquing. They allow to test such systems on a real use case and for highlighting some limitations of these approaches.

### 4.2 Introduction

**Explanations of recommendations are beneficial.** Modern recommender systems accurately capture users' preferences and achieve high performance. But, their performance comes at the cost of increased complexity, which makes them seem like black boxes to users. This may result in distrust or rejection of the recommendations (Herlocker et al., 2000; Tintarev and Masthoff, 2015).

There is thus value in providing *textual explanations* of the recommendations, especially on e-commerce websites, because such explanations enable users to understand why a particular item has been suggested and hence to make better decisions (Chang et al., 2016; Bellini et al., 2018; Tintarev and Masthoff, 2015). Furthermore, explanations increase overall system transparency (Tintarev and Masthoff, 2015; Sinha and Swearingen, 2002) and trustworthiness (Zhang and Curley, 2018; Kunkel et al., 2018).

However, not all explanations are equivalent. Kunkel et al. (2019); Chang et al. (2016) showed that highly personalized justifications using *natural language* lead to substantial increases in perceived recommendation quality and trustworthiness compared to simpler explanations, such as aspect (Wu et al., 2019a), template (Zhang et al., 2014), or similarity (Herlocker et al., 2000).

A second, and more important, benefit of explanations is that they provide a basis for feedback (Lyu et al., 2021): if a user is unsatisfied with a recommendation, understanding what generated it allows them to *critique* it (Figure 4.1). Critiquing – a conversational method of incorporating user preference feedback regarding item attributes into the recommended list of items – has several advantages. First, it allows the system to correct and improve an incomplete or inaccurate model of the user's preferences (Faltings et al., 2004b), which improves the user's decision accuracy (Pu and Chen, 2005; Chen and Pu, 2012). Compared to preference

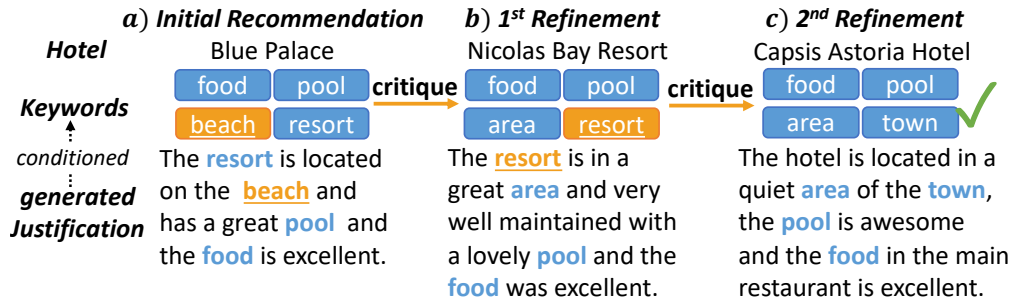


Figure 4.1: A flow of conversational critiquing over two time steps. a) The system proposes to the user a recommendation with a keyphrase explanation and a justification. The user can interact with the explanation and critique phrases. b) A new recommendation is produced from the user’s profile and the critique. c) This process repeats until the user accepts the recommendation and ceases to provide critiques.

elicitation, critiquing is more flexible: users can express preferences in any order and on any criteria (Reilly et al., 2005).

**Useful explanations are hard to generate.** Prior research has employed users’ reviews to capture their preferences and writing styles (e.g., Ni and McAuley, 2018; Li and Tuzhilin, 2019; Dong et al., 2017). From past reviews, they generate *synthetic* ones that serve as personalized *explanations* of ratings given by users. However, many reviews are noisy, because they partly describe experiences or endorsements. It is thus nontrivial to identify meaningful justifications inside reviews. Ni et al. (2019) proposed a pipeline for identifying justifications from reviews and asked humans to annotate them. Chen et al. (2019b, 2020) set the justification as the first sentence. However, these notions of justification were ambiguous, and they assumed that a review contains only one justification.

Recently, Antognini et al. (2021b) solved these shortcomings by introducing a justification extraction system with no prior limits imposed on their number or structure. This is important because a user typically justifies his overall rating with multiple explanations: one for each aspect the user cares about (Musat et al., 2013; Zhang et al., 2014; Musat and Faltings, 2015). The authors showed that there is a connection between faceted ratings and snippets within the reviews: for each subrating, there exists at least one text fragment that alone suffices to make the prediction. They employed a sophisticated attention mechanism to favor long, meaningful word sequences; we call these *markers*. Building upon their study, we show that these *markers* serve to create better user and item profiles and can inform better user-item pair justifications. Figure 4.2 illustrates the pipeline.

**From explanations to critiquing.** To reflect the overlap between the profiles of a user and an item, one can produce a set of keyphrases and then a synthetic justification. The user can correct his profile, captured by the system, by *critiquing* certain aspects he does not like or that are missing or not relevant anymore and obtain a new justification (Figure 4.1). Wu et al. (2019a) introduced a keyphrase-based critiquing method in which attributes are mined from

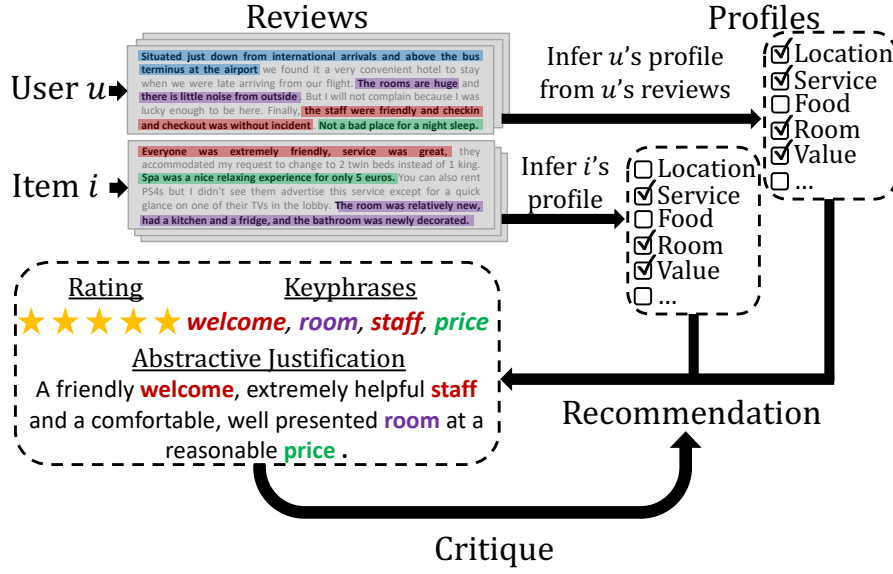


Figure 4.2: For reviews written by a user  $u$  and a set of reviews about an item  $i$ , we extract the justifications for each aspect rating and implicitly build an interest profile. T-RECS outputs a personalized recommendation with two explanations: the keyphrases reflecting the overlap between the two profiles, and a synthetic justification conditioned on the latter.

reviews, and users interact with them. However, their models need an extra autoencoder to project the critique back into the latent space, and it is unclear how the models behave in multi-step critiquing.

We overcome these drawbacks by casting the critiquing as an unsupervised attribute transfer task: altering a keyphrase explanation of a user-item pair representation to the critique. To this end, we entangle the user-item pair with the explanation in the same latent space. At inference, the keyphrase classifier modulates the latent representation until the classifier identifies it as the critique vector.

In this chapter, we address the problem recommendation with fine-grained explanations. We first demonstrate how to extract multiple relevant and personalized justifications from the user's reviews to build a profile that reflects his preferences and writing style (Figure 4.2). Second, we propose T-RECS, a recommender with explanations. T-RECS explains a rating by first inferring a set of keyphrases describing the intersection between the profiles of a user and an item. Conditioned on the keyphrases, the model generates a synthetic personalized justification. We then leverage these explanations in an unsupervised critiquing method for single- and multi-step critiquing. We evaluate our model using two real-world recommendation datasets. T-RECS outperforms strong baselines in explanation generation, effectively re-ranks recommended items in single-step critiquing. Finally, T-RECS also better models the user's preferences in multi-step critiquing while generating consistent textual justifications.

## 4.3 Related Work

### 4.3.1 Textual Explainable Recommendation

Researchers have investigated many approaches to generating textual explanations of recommended items for users. McAuley and Leskovec (2013) proposed a topic model to discover latent factors from reviews and explain recommended items. Zhang et al. (2014) improved the understandability of topic words and aspects by filling template sentences.

Another line of research has generated synthetic reviews as explanations. Prior studies have employed users' reviews and tips to capture their preferences and writing styles. Catherine and Cohen (2017) predicted and explained ratings by encoding the user's review and identifying similar reviews. Lu et al. (2018); Chen et al. (2019b) extended the previous work to generate short synthetic reviews. Sun et al. (2020) optimized both tasks in dual forms. Dong et al. (2017); Costa et al. (2018) proposed an attribute-to-sequence model to learn how to generate reviews given categorical attributes. Ni and McAuley (2018) improved review generation by leveraging aspect information using a sequence-to-sequence model with attention. Instead of reviews, others have generated tips (Li et al., 2017, 2019). However, the tips are scarce and uninformative (Chen et al., 2019b); many reviews are noisy because they describe partially general experiences or endorsements (Ni et al., 2019).

Ni et al. (2019) built a sequence-to-sequence model conditioned on the aspects to generate relevant explanations for an existing recommender system; the fine-grained aspects are provided by the user in the inference. They identified justifications from reviews by segmenting them into elementary discourse units (EDU) (Mann and Thompson, 1988) and asking annotators to label them as "good" or "bad" justifications. Chen et al. (2019b) set the justification as the first sentence. All assumed that a review contains only one justification. Whereas their notions of justification were ambiguous, we extract multiple justifications from reviews using *markers* that justify subratings. Unlike their models, ours predicts keyphrases on which the justifications are conditioned and integrates critiquing.

### 4.3.2 Critiquing

Refining recommended items allows users to interact with the system until they are satisfied. Some methods are example critiquing (Williams and Tou, 1982), in which users critique a set of items; unit critiquing (Burke et al., 1996), in which users critique an item's attribute and request another one instead; and compound critiquing (Reilly et al., 2005) for more aspects. McCarthy et al. (2010) collaboratively utilized critiques from users. The major drawback of these approaches is the assumption of a fixed set of known attributes.

Wu et al. (2019a) circumvented this limitation by extending the neural collaborative filtering model (He et al., 2017). First, the model explains a recommendation by predicting a set of keywords (mined from users' reviews). In Chen et al. (2020), based on Chen et al. (2019b),

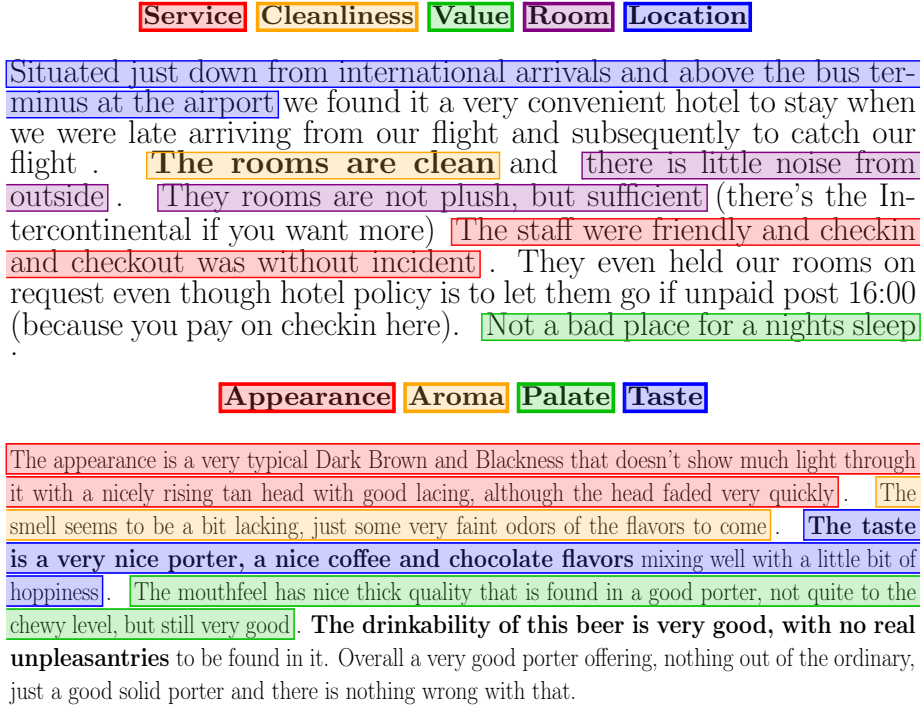


Figure 4.3: Extracted justifications from a hotel review and a beer review. The inferred *markers* depict the excerpts that explain the ratings of the aspects: Service, Cleanliness, Value, Room, and Location. We denote in **bold** the EDU-based justification from the model of (Ni et al., 2019).

the model samples only one keyword via the Gumbel-Softmax function. Our work applies a deterministic strategy similar to Wu et al. (2019a).

Second, Wu et al. (2019a) project the critiqued keyphrase explanations back into the latent space, via an autoencoder that perturbs the training, from which the rating and the explanation are predicted. In this manner, the user's critique modulates his latent representation. The model of Chen et al. (2020) is trained in a two-stage manner: one to perform recommendation and predict one keyword and another to learn critiquing from online feedback, which requires additional data. By contrast, our model is simpler and learns critiquing in an unsupervised fashion: it iteratively edits the latent representation until the new explanation matches the critique. Finally, Luo et al. (2020a) examined various linear aggregation methods on latent representations for multi-step critiquing. In comparison, our gradient-based critiquing iteratively updates the latent representation for each critique.

## 4.4 Extracting Justifications from Reviews

In this section, we introduce the pipeline for extracting high-quality and personalized justifications from users' reviews. We claim that a user justifies his overall experience with multiple



## 4.5 T-RECS: A Multi-Task Transformer with Explanations and Critiquing

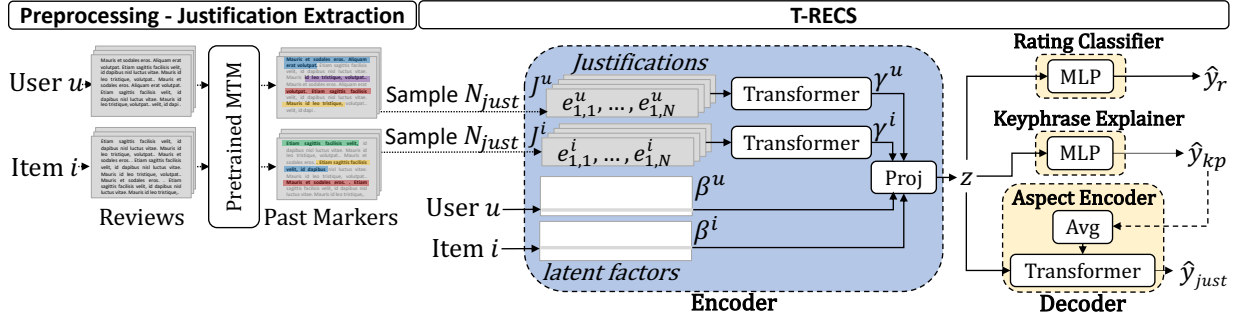


Figure 4.4: (Left) Preprocessing for the users and the items. For each user  $u$  and item  $i$ , we first extract *markers* from their past reviews (highlighted in color), using the pretrained multi-target masker (see Section 4.4), that become their respective justifications. Then, we sample  $N_{just}$  of them and build the justification references  $J^u$  and  $J^i$ , respectively. (Right) T-RECS architecture. Given a user  $u$  and an item  $i$  with their justification references  $J^u, J^i$  and latent factors  $\beta^u, \beta^i$ , T-RECS produces a joint embedding  $z$  from which it predicts a rating  $\hat{y}_r$ , a keyphrase explanation  $\hat{y}_{kp}$ , and a natural language justification  $\hat{y}_{just}$  conditioned on  $\hat{y}_{kp}$ .

explanations: one for each aspect he cares about. Indeed, it has been shown that users write opinions about the topics they care about (Zhang et al., 2014; Musat and Faltings, 2015). Thus, the pipeline must satisfy two requirements:

1. extract text snippets that reflect a rating or subrating, and
2. be data driven and scalable to mine massive review corpora and to construct a large personalized recommendation justification dataset.

In Chapter 2, we proposed the multi-target masker (MTM) to find text fragments that explain faceted ratings in an unsupervised manner. MTM fulfills the two requirements. For each word, the model computes a distribution over the aspect set, which corresponds to the aspect ratings (e.g., service, location) and “not aspect.” In parallel, the model minimizes the number of selected words and discourages aspect transition between consecutive words. These two constraints guide the model to produce long, meaningful sequences of words called **markers**. The model updates its parameters by using the inferred *markers* to predict the aspect sentiments jointly and improves the quality of the *markers* until convergence.

Given a review, MTM extracts the *markers* of each aspect. A sample is shown in Figure 4.3. Similarly to Ni et al. (2019), we filter out *markers* that are unlikely to be suitable justifications: including third-person pronouns or being too short. We use the constituency parse tree to ensure that *markers* are verb phrases. The detailed processing is available in Appendix B.1.

## 4.5 T-RECS: A Multi-Task Transformer with Explanations and Critiquing

Figure 4.4 depicts the pipeline and our proposed T-RECS model. Let  $U$  and  $I$  be the user and item sets. For each user  $u \in U$  (respectively an item  $i \in I$ ), we extract *markers* from the user’s reviews on the training set, randomly select  $N_{just}$ , and build a justification reference  $J^u$  (symmetrically  $J^i$ ).

Given a user  $u$ , an item  $i$ , and their justification history  $J^u$  and  $J^i$ , our goal is to predict 1. a rating  $\mathbf{y}_r$ , 2. a keyphrase explanation  $\mathbf{y}_{kp}$  describing the relationship between  $u$  and  $i$ , and 3. a natural language justification  $\mathbf{y}_{just} = \{w_1, \dots, w_N\}$ , where  $N$  is the length of the justification.  $\mathbf{y}_{just}$  explains the rating  $\mathbf{y}_r$  conditioned on  $\mathbf{y}_{kp}$ .

### 4.5.1 Model Overview

For each user and item, we extract *markers* from their past reviews (in the train set) and build their justification history  $J^u$  and  $J^i$ , respectively (see Section 4.4). T-RECS is divided into four submodels: an **Encoder**  $E$ , which produces the latent representation  $\mathbf{z}$  from the historical justifications and latent factors of the user  $u$  and the item  $i$ ; a **Rating Classifier**  $C^r$ , which classifies the rating  $\hat{\mathbf{y}}_r$ ; a **Keyphrase Explainer**  $C^{kp}$ , which predicts the keyphrase explanation  $\hat{\mathbf{y}}_{kp}$  of the latent representation  $\mathbf{z}$ ; and a **Decoder**  $D$ , which decodes the justification  $\hat{\mathbf{y}}_{just}$  from  $\mathbf{z}$  conditioned on  $\hat{\mathbf{y}}_{kp}$ , encoded via the **Aspect Encoder**  $A$ . T-RECS involves four functions:

$$\mathbf{z} = E(u, i), \quad \hat{\mathbf{y}}_r = C^r(\mathbf{z}), \quad \hat{\mathbf{y}}_{kp} = C^{kp}(\mathbf{z}), \quad \hat{\mathbf{y}}_{just} = D(\mathbf{z}, A(\hat{\mathbf{y}}_{kp})). \quad (4.1)$$

The above formulation contains two types of personalized explanations: a list of keyphrases  $\hat{\mathbf{y}}_{kp}$  that reflects the different aspects of item  $i$  that the user  $u$  cares about (i.e., the overlap between their profiles) and a natural language explanation  $\hat{\mathbf{y}}_{just}$  that justifies the rating, conditioned on  $\hat{\mathbf{y}}_{kp}$ . The set of keyphrases is mined from the reviews and reflects the different aspects deemed important by the users. The keyphrases enable an interaction mechanism: users can express agreement or disagreement with respect to one or multiple aspects and hence critique the recommendation.

#### Entangling user-item.

A key objective of T-RECS is to build a powerful latent representation. It accurately captures user and item profiles with their writing styles and entangles the rating, keyphrases, and a natural language justification. Inspired by the superiority of the Transformer (Vaswani et al., 2017) for text generation tasks (Dathathri et al., 2020; Devlin et al., 2019; Radford et al., 2019; Wang et al., 2019), we propose a Transformer-based encoder that learns latent personalized features from users’ and items’ justifications. We first pass each justification  $J_j^u$  (respectively  $J_j^i$ )

through the Transformer to compute the intermediate representations  $\mathbf{h}_j^u$  (respectively  $\mathbf{h}_j^i$ )<sup>1</sup>. We apply a sigmoid function on the representations and average them to get  $\gamma^u$  and  $\gamma^i$ :

$$\gamma^u = \frac{1}{|J^u|} \sum_{j \in J^u} \sigma(\mathbf{h}_j^u), \quad \gamma^i = \frac{1}{|J^i|} \sum_{j \in J^i} \sigma(\mathbf{h}_j^i). \quad (4.2)$$

In parallel, the encoder maps the user  $u$  (item  $i$ ) to the latent factors  $\beta^u$  ( $\beta^i$ ) via an embedding layer. We compute the latent representation  $\mathbf{z}$  by concatenating the latent personalized features and factors and applying a linear projection:  $\mathbf{z} = E(u, i) = W[\gamma^u \parallel \gamma^i \parallel \beta^u \parallel \beta^i] + \mathbf{b}$ , where  $\parallel$  is the concatenation operator, and  $W$  and  $\mathbf{b}$  the projection parameters.

#### Rating Classifier & Keyphrase Explainer.

Our framework classifies the interaction between the user  $u$  and item  $i$  as positive or negative. Moreover, we predict the keyphrases that describe the overlap of their profiles. Both models are a two-layer feedforward neural network with LeakyRelu activation function. Their respective losses are:

$$\begin{aligned} \mathcal{L}_r(C^r(\mathbf{z}), \mathbf{y}_r) &= (\hat{\mathbf{y}}_r - \mathbf{y}_r)^2, \\ \mathcal{L}_{kp}(C^{kp}(\mathbf{z}), \mathbf{y}_{kp}) &= - \sum_{k=1}^{|K|} y_{kp}^k \log \hat{y}_{kp}^k, \end{aligned} \quad (4.3)$$

where  $\mathcal{L}_r$  is the mean square error,  $\mathcal{L}_{kp}$  the binary cross-entropy, and  $K$  the whole set of keyphrases.

#### Justification Generation.

The last component consists of generating the justification. Inspired by “plan-and-write” (Yao et al., 2019), we advance the personalization of the justification by incorporating the keyphrases  $\hat{\mathbf{y}}_{kp}$ . In other words, T-RECS generates a natural language justification conditioned on the 1. user, 2. item, and 3. aspects of the item that the user would consider important. We encode these via the Aspect Encoder  $A$  that takes the average of their word embeddings from the embedding layer in the Transformer. The aspect embedding is denoted by  $\mathbf{a}_{kp}$  and added to the latent representation:  $\tilde{\mathbf{z}} = \mathbf{z} + \mathbf{a}_{kp}$ . Based on  $\tilde{\mathbf{z}}$ , the Transformer decoding block computes the output probability  $\hat{y}_{just}^{t,w}$  for the word  $w$  at time-step  $t$ . We train using teacher-forcing (Williams and Zipser, 1989) and cross-entropy with label smoothing (Szegedy et al., 2016):

$$\mathcal{L}_{just}(D(\mathbf{z}, \mathbf{a}_{kp}), \mathbf{y}_{just}) = - \sum_{t=1}^{|\mathbf{y}_{just}|} CE(y_{just}^{t,w}, \hat{y}_{just}^{t,w}). \quad (4.4)$$

We train T-RECS end-to-end and minimize jointly the loss  $\mathcal{L} = \lambda_r \mathcal{L}_r + \lambda_{kp} \mathcal{L}_{kp} + \lambda_{just} \mathcal{L}_{just}$ , where  $\lambda_r$ ,  $\lambda_{kp}$ , and  $\lambda_{just}$  control the impact of each loss. All objectives share the latent representation  $\mathbf{z}$  and are thus mutually regularized by the function  $E(u, i)$  to limit overfitting by any objective.

<sup>1</sup>We propose a solution to address the case where users do not have written reviews in Appendix B.3.

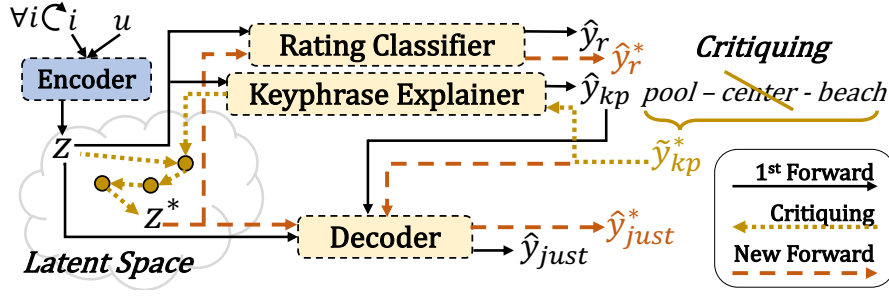


Figure 4.5: Workflow of considering to recommend items to a user  $u$ . We illustrate it for a given item  $i$ . **Black** denotes the forward pass to infer the rating  $\hat{y}_r$  with the explanations  $\hat{y}_{kp}$  and  $\hat{y}_{just}$ . **Yellow** indicates the critiquing: the user critiques the binary-vector keyphrase explanation  $\hat{y}_{kp}$  (e.g., *center*) to  $\tilde{y}_{kp}^*$ , which modulates the latent space into  $\mathbf{z}^*$  for each item. **Orange** shows the new forward pass for the subsequent recommendation  $\hat{y}_r^*$  and explanations  $\hat{y}_{kp}^*, \hat{y}_{just}^*$ .

#### 4.5.2 Unsupervised Critiquing

The purpose of critiquing is to refine the recommendation based on the user's interaction with the explanation, the keyphrases  $\hat{y}_{kp}$ , represented with a binary vector. The user critiques either a keyphrase  $k$  by setting  $\hat{y}_{kp}^k = 0$  (i.e., disagreement) or symmetrically adding a new one (i.e.,  $\hat{y}_{kp}^k = 1$ ). We denote the critiqued keyphrase explanation as  $\tilde{y}_{kp}^*$ .

---

##### Algorithm 1 Iterative Critiquing Gradient Update.

---

- 1: **function** CRITIQUE(latent vector  $\mathbf{z}$ , target keyphrases  $\tilde{y}_{kp}^*$ , trained keyphrases Explainer  $C^{kp}$ , decay coefficient  $\zeta$ , threshold  $T$ , a maximum number of iterations  $I$ )
  - 2:   Set  $\mathbf{z}_0^* = \mathbf{z}$ ,  $\alpha_0 = 1.0$ , and  $t = 1$ ;
  - 3:   **while**  $|C^{kp}(\mathbf{z}^*) - \tilde{y}_{kp}^*| > T$  and  $t \leq I$  **do**
  - 4:      $\mathbf{g}_{t-1} = \nabla_{\mathbf{z}_{t-1}^*} \mathcal{L}_{kp}(C^{kp}(\mathbf{z}_{t-1}^*), \tilde{y}_{kp}^*)$ ;
  - 5:      $\mathbf{z}_t^* = \mathbf{z}_{t-1}^* - \alpha_{t-1} \frac{\mathbf{g}_{t-1}}{\|\mathbf{g}_{t-1}\|_2}$ ;
  - 6:      $\alpha_t = \zeta \alpha_{t-1}$ ;
  - 7:      $t = t + 1$ ;
  - 8: **return**  $\mathbf{z}_t^*$ ;
- 

The overall critiquing process is depicted in Figure 4.5 and Algorithm 1. Inspired by the recent success in editing the latent space on the unsupervised text attribute transfer task (Wang et al., 2019; Dathathri et al., 2020), we employ the trained Keyphrase Explainer  $C^{kp}$  and the critiqued explanation  $\tilde{y}_{kp}^*$  to provide the gradient from which we update the latent representation  $\mathbf{z}$  (depicted in **yellow**). More formally, given a latent representation  $\mathbf{z}$  and a binary critique vector  $\tilde{y}_{kp}^*$ , we want to find a new latent representation  $\mathbf{z}^*$  that will produce a new keyphrase explanation close to the critique, such that  $|C^{kp}(\mathbf{z}^*) - \tilde{y}_{kp}^*| \leq T$ , where  $T$  is a threshold. In order to achieve this goal, we iteratively compute the gradient with respect to  $\mathbf{z}$  instead of the model

Table 4.1: Descriptive statistics of the datasets.

Dataset	#Users	#Items	#Inter.	Dens.	KP Cov.	Avg. #KP per		
						Just.	Rev.	User
Hotel	72,603	38 896	2.2M	0.08%	97.66%	2.15	3.79	115
Beer	7,304	8,702	1.2M	2.02%	96.87%	3.72	6.97	1,210

parameters  $C_\theta^{kp}$ . We then modify  $\mathbf{z}$  in the direction of the gradient until we get a new latent representation  $\mathbf{z}^*$  that  $C^{kp}$  considers close enough to  $\tilde{\mathbf{y}}_{kp}^*$  (shown in **orange**). We emphasize that we use the gradient to modulate  $\mathbf{z}$  rather than the parameters  $C^{kp}$ .

Let denote the gradient as  $\mathbf{g}_t$  and a decay coefficient as  $\zeta$ . For each iteration  $t$  and  $\mathbf{z}_0^* = \mathbf{z}$ , the modified latent representation  $\mathbf{z}_t^*$  at the  $t^{\text{th}}$  iteration can be formulated as follows:

$$\mathbf{g}_t = \nabla_{\mathbf{z}_t^*} \mathcal{L}_{kp}(C^{kp}(\mathbf{z}_t^*), \tilde{\mathbf{y}}_{kp}^*); \quad \mathbf{z}_t^* = \mathbf{z}_{t-1}^* - \zeta^{t-1} \mathbf{g}_t / \|\mathbf{g}_t\|_2. \quad (4.5)$$

Because this optimization is nonconvex, there is no guarantee that the difference between the critique vector and the inferred explanation will differ by only  $T$ . In our experiments in Section 4.6.5, we found that a limit of 50 iterations works well in practice, and that the newly induced explanations remain consistent.

## 4.6 Experiments

In this section, we answer the following research questions:

- **RQ 1:** Are *markers* appropriate justifications for recommendation?
- **RQ 2:** Does T-RECS generate high-quality, relevant, and personalized explanations?
- **RQ 3:** Do T-RECS justifications benefit the overall recommendation quality?
- **RQ 4:** Can T-RECS enable critiquing and effectively re-rank recommended items by critiquing explanations?

### 4.6.1 Experimental Settings

#### Datasets.

We evaluate the quantitative performance of T-RECS using two real-world, publicly available datasets: BeerAdvocate (McAuley et al., 2012) and HotelRec (Antognini and Faltings, 2020b). They contain 1.5 and 50 million reviews from BeerAdvocate and TripAdvisor. In addition to

the overall rating, users also provided five-star aspect ratings<sup>2</sup>. We binarize the ratings with a threshold  $t$ :  $t > 4$  for hotel reviews and  $t > 3.5$  for beer reviews. We further filter out all users with fewer than 20 interactions and sort them chronologically. We keep the first 80% of interactions per user as the training data, leaving the remaining 20% for validation and testing. We sample two justifications per review. Table 4.1 shows the statistics of the datasets.

We need to select keyphrases for explanations and critiquing. Hence, we follow the processing in Wu et al. (2019a) to extract 200 keyphrases (distributed uniformly over the aspect categories) from the *markers* on each dataset. Some examples are shown in Appendix B.2.

### Implementation Details.

To extract *markers*, we trained MTM with the hyperparameters following Chapter 2. We obtained approximately 79% and 91% macro F1 Scores as shown in Table 2.4. Because the multi-aspect classification is not the focus, here we ignore the training procedure and performance evaluation.

We build the justification history  $J^u, J^i$ , with  $N_{just} = 32$ . We set the embedding and attention dimension to 256 and to 1024 for the feed-forward network. The encoder and decoder consist of two layers of Transformer with 4 attention heads. We use a batch size of 128, dropout of 0.1, and Adam with learning rate 0.001. For critiquing, we choose a threshold and decay coefficient  $T = 0.015, \zeta = 0.9$  and  $T = 0.01, \zeta = 0.975$  for hotel and beer reviews. We tune all models on the dev set. For reproducibility purposes, we provide details in Appendix F.3.

#### 4.6.2 RQ 1: Are *markers* appropriate justifications?

We first verify whether *markers* can serve as justifications for recommendation. We derive baselines from Ni et al. (2019): we split a review into elementary discourse units (EDUs) and apply their classifier to get justifications; it is trained on a manually annotated dataset and generalizes well to other domains. We employ two variants: EDU One and EDU All. The latter includes all justifications, whereas the former includes only one.

We perform a human evaluation using Amazon’s Mechanical Turk (see Appendix D.2 for more details) to judge the quality of the justifications extracted from the Markers, EDU One, and EDU All on both datasets. We employ three setups: an evaluator is presented with 1. the three types of justifications; 2. only those from Markers and EDU All; and 3. EDU One instead of EDU All. We sampled 300 reviews (100 per setup) with generated justifications presented in random order. The annotators judged the justifications by choosing the most convincing in the pairwise setups and otherwise using best-worst scaling (Louviere et al., 2015), which has been shown to give more reliable results than Likert scales (van der Lee et al., 2019; Kiritchenko and Mohammad, 2016). We report the win rates for the pairwise comparisons and a normalized

---

<sup>2</sup>In cases where faceted ratings are not available at large cases, Antognini and Faltings (2021b); Mukherjee and Awadallah (2020); Niu et al. (2020) proposed elegant solutions to infer them from 20 or fewer samples.

Table 4.2: Human evaluation of explanations in terms of the win rate and the best-worst scaling. A score significantly different than Markers (post hoc Tukey HSD test) is denoted by \*\* for  $p < 0.001$ . Human annotators significantly prefer Markers as explanations.

		<i>Hotel</i>			<i>Beer</i>		
	Winner Loser	Win Rate			Win Rate		
	Markers EDU All	81%**			77%**		
	Markers EDU One	93%**			90%**		

Model	<i>Hotel</i>			<i>Beer</i>		
	Score	#B	#W	Score	#B	#W
EDU One	-0.95**	1	96	-0.93**	2	95
EDU All	0.21**	24	3	0.20**	23	3
Markers (Ours)	<b>0.74</b>	75	1	<b>0.73</b>	75	2

score ranging from -1 to +1.

Table 4.2 shows that justifications extracted from Markers are preferred, on both datasets, more than 80% of the time. Moreover, when compared to EDU All and EDU One, Markers achieve a score of 0.74, three times higher than EDU All. Therefore, justifications extracted from the Markers are significantly better than EDUs, and a single justification cannot explain a review. Figure 4.3 shows a sample for comparison.

#### 4.6.3 RQ 2: Does T-RECS infer high-quality and personalized explanations?

We investigate whether T-RECS can generate *personalized* explanations for a given user-item pair: 1. a relevant natural language justifications and 2. a list of keyphrases, that best describes the intersection of the user and item profiles.

##### Natural Language Explanations.

We consider five baselines: ExpansionNet (Ni and McAuley, 2018) is a sequence-to-sequence model with a user, item, aspect, and fusion attention mechanism that generates personalized reviews. DualPC (Sun et al., 2020) and CAML (Chen et al., 2019b) generate an explanation based on a rating and the user-item pair. Ref2Seq improves upon ExpansionNet by learning only from historical justifications of a user and an item. AP-Ref2Seq (Ni et al., 2019) extends Ref2Seq with aspect planning (Yao et al., 2019), in which aspects are given during the generation. All models use beam search ( $k = 3$ ) during testing and the same keyphrases as aspects. For automatic evaluation, we employ BLEU (Papineni et al., 2002), ROUGE-L (Lin and Hovy, 2002), and BertScore (Zhang et al., 2020), a similarity metric based on BERT embeddings that has been shown to correlate better with human judgments. We also report the perplexity for evaluating the fluency and  $R_{KW}$  for the explanation consistency as in Chen et al. (2020): the

Table 4.3: Generated justifications on automatic evaluation using language generation metrics.

	Model	BLEU	R-L	BERT <sub>Score</sub>	PPL↓	R <sub>KW</sub>
<i>Hotel</i>	ExpansionNet	0.53	6.91	74.81	28.87	60.09
	DualPC	1.53	16.73	86.76	28.99	13.12
	CAML	1.13	16.67	87.77	29.10	9.38
	Ref2Seq	1.77	16.45	86.74	29.07	13.19
	AP-Ref2Seq	7.28	33.71	88.31	21.31	90.20
	T-RECS (Ours)	<b>7.47</b>	<b>34.10</b>	<b>90.23</b>	<b>17.80</b>	<b>93.57</b>
<i>Beer</i>	ExpansionNet	1.22	9.68	72.32	22.28	82.49
	DualPC	2.08	14.68	85.49	21.15	10.60
	CAML	2.43	14.99	85.96	21.29	10.18
	Ref2Seq	3.51	15.96	85.27	22.34	12.10
	AP-Ref2Seq	15.89	46.50	91.35	12.07	91.52
	T-RECS (Ours)	<b>16.54</b>	<b>47.20</b>	<b>91.50</b>	<b>10.24</b>	<b>94.96</b>

Table 4.4: Human evaluation of justifications in terms of best-worst scaling for **Overall**, **Fluency**, **Informativeness**, and **Relevance**. Most scores are significantly different than T-RECS (post hoc Tukey HSD test) with  $p < 0.002$ . † denotes a nonsignificant score. Explanations generated by T-RECS are largely preferred.

	<i>Hotel</i>				<i>Beer</i>			
Model	O	F	I	R	O	F	I	R
ExpansionNet	-0.58	-0.67	-0.52	-0.56	-0.03	-0.31	0.10	-0.01
Ref2Seq	-0.27	-0.19	-0.30	-0.26	-0.69	-0.34	-0.71	-0.69
AP-Ref2Seq	0.30	0.32	0.29	0.29	0.22	0.25†	0.21†	0.25
T-RECS (Ours)	<b>0.55</b>	<b>0.54</b>	<b>0.53</b>	<b>0.53</b>	<b>0.49</b>	<b>0.39</b>	<b>0.39</b>	<b>0.45</b>

ratio of the target keyphrases present in the generated justifications.

The main results are presented in Table 4.3 (more in Appendix A.2). T-RECS achieves the highest scores on both datasets. We note that 1. sequence-to-sequence models better capture user and item information to produce more relevant justifications, and 2. using a keyphrase plan doubles the performance on average and improving explanation consistency.

Because the justifications are intended for end users, we thus conduct a human evaluation, with the best models according to  $R_{KW}$ , on four dimensions using best-worst scaling:

1. **Overall** measures the overall subjective quality;
2. **Fluency** represents the structure, grammar, and readability;
3. **Informativeness** indicates whether the justification contains information pertinent to the user;
4. **Relevance** measures how relevant the information is to an item.

We sample 300 explanations and showed them in random order.



Table 4.5: Comparisons of the extracted justifications from different models for two hotels on the hotel dataset. Colors denote aspects while underline denotes EDUs classified as “good” justifications.

Model	Casa del Sol Machupicchu	Southern Sun Waterfront Cape Town
Review	<p><u>the hotel was decent</u>      <u>the</u>      <u>staff</u>  <u>was very friendly.</u>      the free pisco  sour class with kevin was a nice bonus! however,  the rooms were lacking. <u>the wifi was incredibly</u>  <u>slow and there was no air conditioning, so it got</u>  <u>very hot at night.</u> we couldn't open the windows  either because there were so many bugs, birds,  and noise. <u>overall, the location is convenient, but</u>  <u>was is not worth the price.</u></p>	<p>this is my second year visit-  ing cape town and staying here.  <u>excellent location to business district, conven-</u>  <u>tion center, v&amp;a waterfront and access short</u>  <u>distance to table mountain.</u>      <u>very nice</u>  <u>hotel, very friendly staff. breakfast is very good.</u>  <u>rooms are nice</u> but bed mattress could  be improved as bed is somewhat hard.  <u>overall a very nice hotel.</u></p>
Rating	<p>Overall: 3.0, <u>Service: 3.0</u>, <u>Cleanliness: 4.0</u>, <u>Value:</u>  <u>2.0</u>, <u>Location: 4.0</u>, Room: 3.0</p>	<p>Overall: 4.0, <u>Service: 5.0</u>, <u>Cleanliness: 5.0</u>,  <u>Value: 4.0</u>, <u>Location: 5.0</u>, Room: 3.0</p>
Markers	<ul style="list-style-type: none"> <li>- the rooms were lacking.</li> <li>- <u>the hotel was decent and the staff was very friendly.</u></li> <li>- <u>overall , the location is convenient , but was is not worth the price.</u></li> <li>- <u>we could n't open the windows either because there were so many bugs , birds , and noise.</u></li> <li>- <u>the wifi was incredibly slow and there was no air conditioning , so it got very hot at night.</u></li> </ul>	<ul style="list-style-type: none"> <li>- <u>breakfast is very good.</u></li> <li>- <u>very nice hotel , very friendly staff.</u></li> <li>- <u>rooms are nice but bed mattress could be improved as bed is somewhat hard.</u></li> <li>- <u>excellent location to business district , conven-</u>  <u>tion center , v&amp;a waterfront and access short</u>  <u>distance to table mountain.</u></li> </ul>
EDU All	<ul style="list-style-type: none"> <li>- <u>the hotel was decent</u></li> <li>- <u>the free pisco sour class with kevin was a nice bonus.</u></li> </ul>	<ul style="list-style-type: none"> <li>- <u>excellent location to business district , conven-</u>  <u>tion center , v&amp;a waterfront and access short</u>  <u>distance to table mountain.</u></li> <li>- <u>very nice hotel , very friendly staff. breakfast is</u>  <u>very good.</u></li> <li>- <u>rooms are nice.</u></li> <li>- <u>overall a very nice hotel.</u></li> </ul>
EDU One	<ul style="list-style-type: none"> <li>- the hotel was decent</li> </ul>	<ul style="list-style-type: none"> <li>- very nice hotel , very friendly staff . breakfast is very good</li> </ul>

Table 4.4 shows that our explanations are largely preferred on all criteria. T-RECS outperforms all methods on all criteria and performs 1.8 times better than AP-Ref2Seq. Interestingly, we observe that ExpansionNet achieves significantly better results on the beer dataset with scores around 0.00 for *Overall* and *Relevance*, and 0.10 for *Informativeness*; the information contained in beer reviews is easier to capture, as observed in Bao et al. (2018).

We provide samples of justifications generated by all models in Table 4.5 and Table 4.6. We observe that the *markers* justify the subratings. Although they might be some overlaps between EDU All and Markers, justifications from EDU All often are incomplete or not relevant.

## Chapter 4. Generating Explanations and Interacting with Them through Critiquing

Table 4.6: Comparisons of the extracted justifications from different models for two beers on the beer dataset. Colors denote aspects while underline denotes EDUs classified as “good” justifications.

Model	Saison De Lente	Bell's Porter
Review	poured from a 750ml bottle into a chimay branded chalice. a: <b>cloudy and unfiltered with a nice head that lasts and leaves good amounts of lacing in its tracks.</b> s: <b>sour and bready with apple and yeast hints in there as well.</b> t: <b>dry and hoppy with a nice crisp sour finish.</b> m: <b>medium bodied, high carbonation with big bubbles.</b> d: easy to drink, but i didn't really want more after splitting a 750ml with a buddy of mine.	<b>this beer pours black with a nice big frothy offwhite head.</b> <b>smells or roasted malts, and chocolate.</b> <b>tastes of roasted malt with some chocolate and a hint of coffee.</b> the mouthfeel has medium body and is semi-smooth with some nice carbination. drinkability is decent i could drink a couple. <u>overall a good choice</u> from bell's.
Rating	Overall: 3.0, <b>Appearance: 3.5</b> , <b>Smell: 4.0</b> , <b>Mouthfeel: 3.5</b> , <b>Taste: 3.5</b>	Overall: 3.5, <b>Appearance: 4.0</b> , <b>Smell: 3.5</b> , <b>Mouthfeel: 3.5</b> , <b>Taste: 4.0</b>
Markers	<ul style="list-style-type: none"> <li>- <b>dry and hoppy with a nice crisp sour finish.</b></li> <li>- <b>medium bodied , high carbonation with big bubbles.</b></li> <li>- <b>sour and bready with apple and yeast hints in there as well.</b></li> <li>- <b>cloudy and unfiltered with a nice head that lasts and leaves good amounts of lacing in its tracks.</b></li> </ul>	<ul style="list-style-type: none"> <li>- <b>smells or roasted malts , and chocolate.</b></li> <li>- <b>this beer pours black with a nice big frothy offwhite head.</b></li> <li>- <b>tastes of roasted malt with some chocolate and a hint of coffee.</b></li> <li>- <b>the mouthfeel has medium body and is semi smooth with some nice carbination.</b></li> </ul>
EDU All	<ul style="list-style-type: none"> <li>- <u>medium bodied , high carbonation with big bubbles.</u></li> <li>- <u>easy to drink</u></li> </ul>	<ul style="list-style-type: none"> <li>- <u>smells or roasted malts , and chocolate.</u></li> <li>- <u>tastes of roasted malt with some chocolate and a hint of coffee.</u></li> <li>- drinkability is decent</li> <li>- overall a good choice</li> </ul>
EDU One	<ul style="list-style-type: none"> <li>- easy to drink</li> </ul>	<ul style="list-style-type: none"> <li>- drinkability is decent</li> </ul>

Table 4.7: Keyphrase explanation quality using raking metrics at  $N = 10$ .

Model	<i>Hotel</i>				<i>Beer</i>			
	NDCG	MAP	P	R	NDCG	MAP	P	R
Pop	0.333	0.208	0.143	0.396	0.250	0.229	0.176	0.253
E-NCF	0.341	0.215	0.137	0.380	0.249	0.220	0.179	0.262
CE-NCF	0.229	0.143	0.092	0.255	0.192	0.172	0.136	0.197
E-VNCF	0.344	0.216	0.139	0.386	0.236	0.210	0.170	0.248
CE-VNCF	0.229	0.134	0.107	0.297	0.203	0.178	0.148	0.215
T-RECS (Ours)	<b>0.376</b>	<b>0.236</b>	<b>0.158</b>	<b>0.436</b>	<b>0.316</b>	<b>0.280</b>	<b>0.228</b>	<b>0.332</b>

### Keyphrase Explanations.

Predicting keyphrases from the user-item latent representation is a natural way to entangle them with and enable critiquing (see Section 4.5). We compare T-RECS with the popularity baseline and the models proposed in Wu et al. (2019a), which are extended versions of the NCF model (He et al., 2017). E-NCF and CE-NCF augment the NCF method with an explanation and a critiquing neural component. Also, the authors provide variational variants: VNCF, E-VNCF, and CE-VNCF. Here, we omit NCF and VNCF because they are trained only to predict ratings. We report the following metrics: NDCG, MAP, Precision, and Recall at 10 (see Appendix A.3 for more metrics).

Table 4.7 shows that T-RECS outperforms the CE-(V)NCF models by 60%, Pop by 20%, and E-(V)NCF models by 10% to 30% on all datasets. Interestingly, Pop performs better than CE-(V)NCF, showing that many keywords are recurrent in reviews. Although the task is thus not trivial, T-RECS retrieves up to 60% of relevant keyphrases within the top 20. Thus, predicting keyphrases from the user-item latent space is a natural way to entangle them with (and enable critiquing).

#### 4.6.4 RQ 3: Do T-RECS justifications benefit the overall recommendation quality?

In this section, we investigate whether justifications are beneficial to T-RECS and improve overall performance. We assess the performance on three different axes: rating prediction, preference prediction, and Top-N recommendation.

#### Rating & Preference Prediction.

We first analyze recommendation performance by the mean of rating prediction. We utilize the common Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics. However, the rating prediction performance alone does not best reflect the quality of recommendations, because users mainly see the relative ranking of different items (Ricci et al., 2011; Musat and

Table 4.8: Performance of the rating prediction. Kendall’s  $\tau$  denotes the correlation between the users’ own rankings and the models’ ones. T-RECS consistently produces better recommendation.

Model	<i>Hotel</i>			<i>Beer</i>		
	MAE	RMSE	$\tau \uparrow$	MAE	RMSE	$\tau \uparrow$
NMF	0.3825	0.6171	0.2026	0.3885	0.4459	0.4152
PMF	0.3860	0.5855	0.0761	0.3922	0.4512	0.4023
HFT	0.3659	0.4515	0.4584	0.3616	0.4358	0.4773
NARRE	0.3564	0.4431	0.4476	0.3620	0.4377	0.4506
NCF	0.3619	0.4358	0.4200	0.3638	0.4341	0.4696
E-NCF	0.3579	0.4382	0.4145	0.3691	0.4326	0.4685
CE-NCF	0.3552	0.4389	0.4165	0.3663	0.4390	0.4527
VNCF	0.3502	0.4313	0.4408	0.3666	0.4300	0.4706
E-VNCF	0.3494	0.4365	0.4072	0.3627	0.4457	0.4651
CE-VNCF	0.3566	0.4545	0.3502	<b>0.3614</b>	0.4330	0.4619
T-RECS (Ours)	<b>0.3306</b>	<b>0.4305</b>	<b>0.4702</b>	<b>0.3614</b>	<b>0.4295</b>	<b>0.4909</b>

Faltings, 2015). Consequently, we measure also how well the item rankings computed by T-RECS agree with the user’s own rankings as given by his own review ratings. We measure this quality by leveraging the standard metric Kendall’s  $\tau$  rank correlation (Kendall, 1938), computed on all pairs of rated-items by a user in the testing set. Overall, there are 153,954 and 1,769,421 pairs for the hotel and beer datasets, respectively.

We examine the following baseline methods together with T-RECS: NMF (Hoyer, 2004) is a non-negative matrix factorization model for rating prediction. PMF (Mnih and Salakhutdinov, 2008) is a probabilistic matrix factorization method using ratings for collaborative filtering. HFT (McAuley and Leskovec, 2013) is a strong latent-factor baseline, combined with a topic model aiming to find topics in the review text that correlate with the users’ and items’ latent factors. NARRE (Chen et al., 2018a) is a state-of-the-art model that predicts ratings and reviews’ usefulness jointly. Finally, we include the six methods of Wu et al. (2019a) described in Section 4.6.3.

The results are shown in Table 4.8. T-RECS consistently outperforms all the baselines, by a wide margin on the hotel dataset, including models based on collaborative filtering with/without review information or models extended with an explanation component and/or a critiquing component. Interestingly, the improvement in the hotel dataset in terms of MAE and RMSE is significantly higher than on the beer dataset. We hypothesize that this behavior is due to the sparsity (see Table 4.1), which has also been observed in the hotel domain in prior work (Musat and Faltings, 2015; Antognini and Faltings, 2020b). On the beer dataset, we note that reviews contain strong indicators and considerably improve the performance of NARRE and HFT compared to collaborative filtering methods. The extended (V)NCF models with

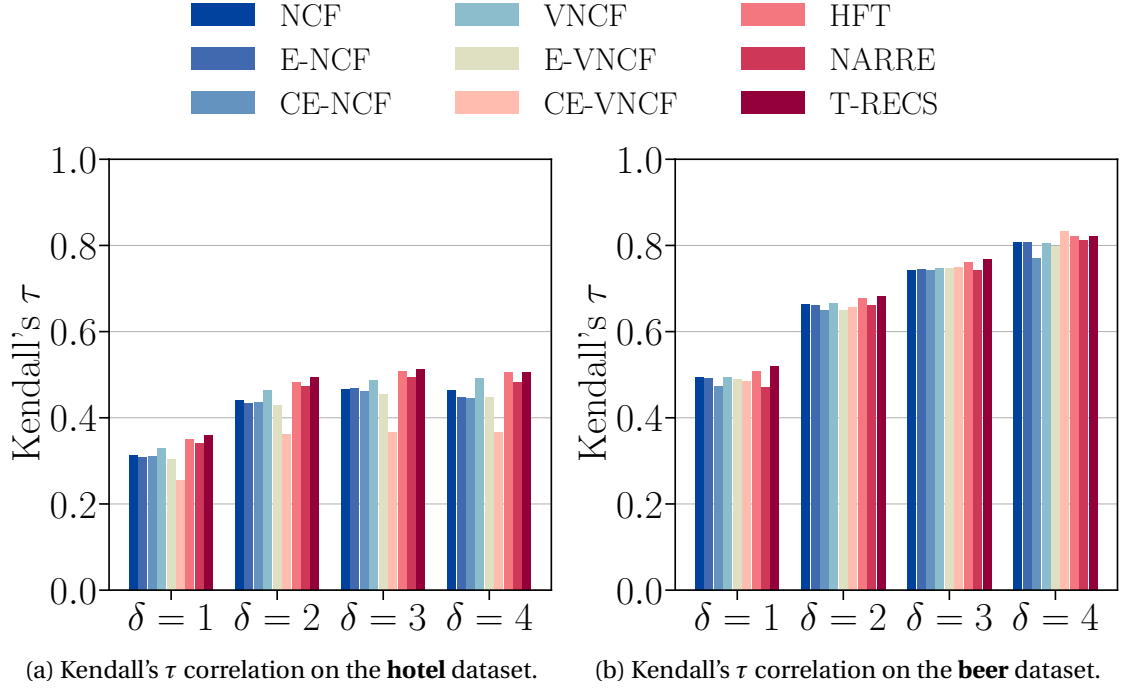


Figure 4.6: Performance of the preference prediction using Kendall's  $\tau$  and  $\delta = |y_r^i - y_r^j|$ .

either an explanation and/or a critiquing component improve MAE performance. Therefore, explanations can benefit the recommender systems to improve rating prediction.

Table 4.8 also contains the results in terms of preference prediction. T-RECS achieves up to 0.0136 (+2.85%) higher Kendall correlation compared to the best baseline. Surprisingly, we note that CE-VNCF, NMF, and PMF show much worse results on the hotel datasets than on the beer dataset. This highlights that hotel reviews are noisier than beer reviews and emphasizes the importance of capturing users' profiles, where T-RECS does best in comparison to other models.

### Preference Prediction.

In this experiment, we study a more fine-grained rank correlation. Following Musat et al. (2013); Musat and Faltings (2015), we analyze the pairwise ranking of rated items by a user, and we impose a minimum value for the rating difference between two items  $i$  and  $j$ , such that  $\delta = |y_r^i - y_r^j|$ ; the rating difference  $\delta$  symbolizes the minimum preference strength.

Figure 4.6 contains the Kendall's  $\tau$  evaluation for multiple  $\delta$  on both datasets. Overall, T-RECS increases the Kendall correlation similarly to other models but performs better on average. We observe that HFT's performance is similar to T-RECS, although slightly lower for most cases. On the beer dataset, we surprisingly note that CE-VNCF obtains a negligible higher score for  $\delta = 4$ , while significantly underperforming for  $\delta < 4$ , and especially on the hotel dataset.

Table 4.9: Performance of the Top- $N$  recommendation.

	Model	NDCG@N		Precision@N		Recall@N	
		N=10	N=20	N=10	N=20	N=10	N=20
<i>Hotel</i>	NCF	0.1590	0.2461	0.0231	0.0200	0.2310	0.3991
	E-NCF	0.1579	0.2432	0.0234	0.0200	0.2336	0.4004
	CE-NCF	0.1585	0.2431	0.0235	0.0201	0.2352	0.4028
	VNCF	0.1492	0.2431	0.0220	0.0197	0.2204	0.3932
	E-VNCF	0.1505	0.2395	0.0219	0.0192	0.2188	0.3842
	CE-VNCF	<b>0.1738</b>	<b>0.2662</b>	0.0221	0.0190	0.2210	0.3809
	T-RECS (Ours)	0.1674	<b>0.2662</b>	<b>0.0236</b>	<b>0.0207</b>	<b>0.2358</b>	<b>0.4144</b>
<i>Beer</i>	NCF	0.2172	0.3509	0.0250	0.0212	0.2499	0.4231
	E-NCF	0.2087	0.3363	0.0243	0.0205	0.2426	0.4103
	CE-NCF	0.2226	0.3456	0.0252	0.0205	0.2517	0.4105
	VNCF	0.1943	0.3329	0.0235	0.0211	0.2345	0.4213
	E-VNCF	0.1387	0.2813	0.0158	0.0168	0.1579	0.3362
	CE-VNCF	0.2295	0.3598	0.0263	0.0218	0.2630	0.4352
	T-RECS (Ours)	<b>0.2372</b>	<b>0.3674</b>	<b>0.0269</b>	<b>0.0219</b>	<b>0.2693</b>	<b>0.4390</b>

Finally, the Kendall's  $\tau$  correlation increases majorly with the strength of preference pairs on the beer dataset and plateaus over  $\delta \geq 2$  on the hotel dataset. It highlights that hotel reviews are noisier than beer reviews, and it emphasizes the importance of capturing users' profiles, where T-RECS does best in comparison to other models.

#### Recommendation Performance.

We evaluate the performance of T-RECS on the last dimension: Top- $N$  recommendation. We adopt the widely used leave-one-out evaluation protocol (Nikolakopoulos et al., 2019; Zhao et al., 2018); in particular, for each user, we randomly select one liked item in the test set alongside 99 randomly selected unseen items. We compare T-RECS with the state-of-the-art methods in Wu et al. (2019a). Finally, we rank the item lists based on the recommendation scores produced by each method, and report the NDCG, Precision, and Recall at different  $N$ .

Table 4.9 presents the main results. Comparing to CE-(V)NCF models, which contain an explanation and critiquing components, our proposed model shows better recommendation performance for almost all metrics on the two datasets. On average, the variants (V)NCF reach higher results than the original method, which was not the case in the rating prediction and relative rankings tasks (see Section 4.6.4), unlike T-RECS that shows consistent results.

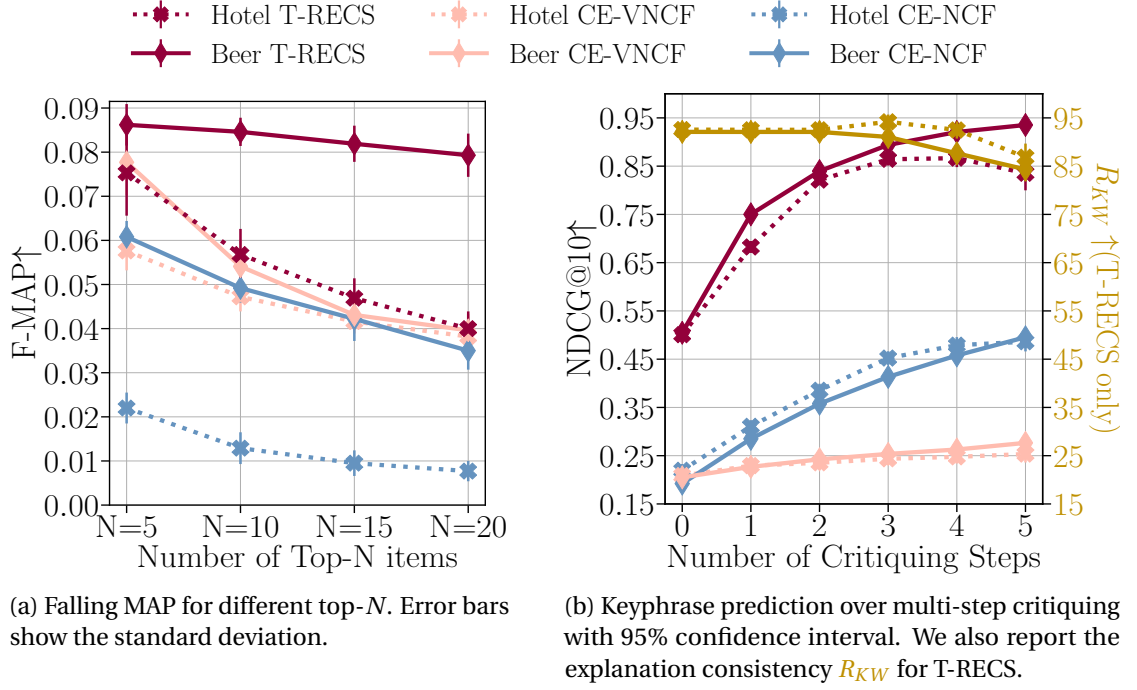


Figure 4.7: Single- (top) and multi-step (bottom) critiquing.

#### 4.6.5 RQ 4: Can T-RECS enable critiquing?

We now investigate whether T-RECS can fill the gap between justifications and recommendation by enabling critiquing and effectively re-ranking recommended items.

##### Single-Step Critiquing.

For a given user, T-RECS recommends an item and generates personalized explanations, where the user can interact by critiquing one or multiple keyphrases. However, no explicit ground truth exists to evaluate the critiquing. We use F-MAP (Wu et al., 2019a) to measure the effect of a critique. Given a user, a set of recommended items  $\mathbb{S}$ , and a critique  $k$ , let  $\mathbb{S}_k$  be the set of items containing  $k$  in the explanation. The F-MAP measures the ranking difference of the affected items  $\mathbb{S}_k$  before and after critiquing  $k$ , using the Mean Average Precision at  $N$ :

$$\text{F-MAP} = \text{MAP}@N_{\mathbb{S}_k}^{\text{before}} - \text{MAP}@N_{\mathbb{S}_k}^{\text{after}}. \quad (4.6)$$

A positive F-MAP indicates that the rank of items in  $\mathbb{S}_k$  fell after  $k$  is critiqued. We compare T-RECS with CE-(V)NCF and average the F-MAP over 5,000 user-keyphrase pairs.

Figure 4.7a presents the F-MAP performance on both datasets. All models show an anticipated positive F-MAP. The performance of T-RECS improves considerably on the beer dataset and is significantly higher for  $N \leq 10$  on the hotel dataset. T-RECS and CE-(V)NCF handle the critique in different ways. CE-(V)NCF use an autoencoder learned jointly during training

that projects the keyphrase explanation with the critiqued keyphrase removed back into the latent space. In contrast, T-RECS edits the entangled latent representation conforming to the new target explanation in the direction of the Keyphrase Explainer gradient. The gap in performance may be caused by the extra loss of the autoencoder, which brings noise during training. T-RECS only iteratively edits the latent representation at test time.

### Multi-Step Critiquing.

Evaluating multi-step critiquing via ranking is difficult because many items can have the keyphrases of the desired item. Instead, we evaluate whether a system obtains a complete model of the user's preferences following Pu et al. (2006). To do so, a simulated user expresses his keyphrase preferences iteratively according to a randomly selected liked item. After each newly state preference, we evaluate the keyphrase explanations in the same manner as in Section 4.6.3 (we report complementary metrics in Appendix A.4). For T-RECS, we also report the explanation consistency  $R_{KW}$ . We expect a model to improve its performance over critiquing.

We run up to five-steps critiques over 1,000 random selected users and up to 5,000 random keyphrases for each dataset. Figure 4.7b shows that T-RECS builds through the critiques more accurate user profiles and consistent explanations. CE-NCF's top performance is significantly lower than T-RECS, and CE-VNCF plateaus, surely because of the KL divergence regularization, which limits the amount of information stored in the latent space. The explanation quality in T-RECS depends on the accuracy of the user's profile and may become saturated once we find it after four steps.

## 4.7 Multi-Step Critiquing User Interface

In this section, we introduce four different web interfaces<sup>3</sup> helping users make decisions and find their ideal item. We have chosen the hotel recommendation domain as a use case even though our approach is trivially adaptable for other domains. Our system is model-agnostic (for both the recommender and critiquing models), allowing great flexibility and further extensions. Our interfaces are, above all, a useful tool to help research in recommendation with standard critiquing and critiquing based on neural generated-text techniques on a real use case. Finally, they can help to highlight the limitations of these systems to find ways to overcome them; they are not intended to be a finished usable product for end consumers.

### 4.7.1 System Overview

The system overview is shown in Figure 4.8a. To illustrate the flexibility of our system, we employ the HotelRec dataset (Antognini and Faltings, 2020b), which contains 50 million

---

<sup>3</sup>Available under <https://lia.epfl.ch/critiquing/>.



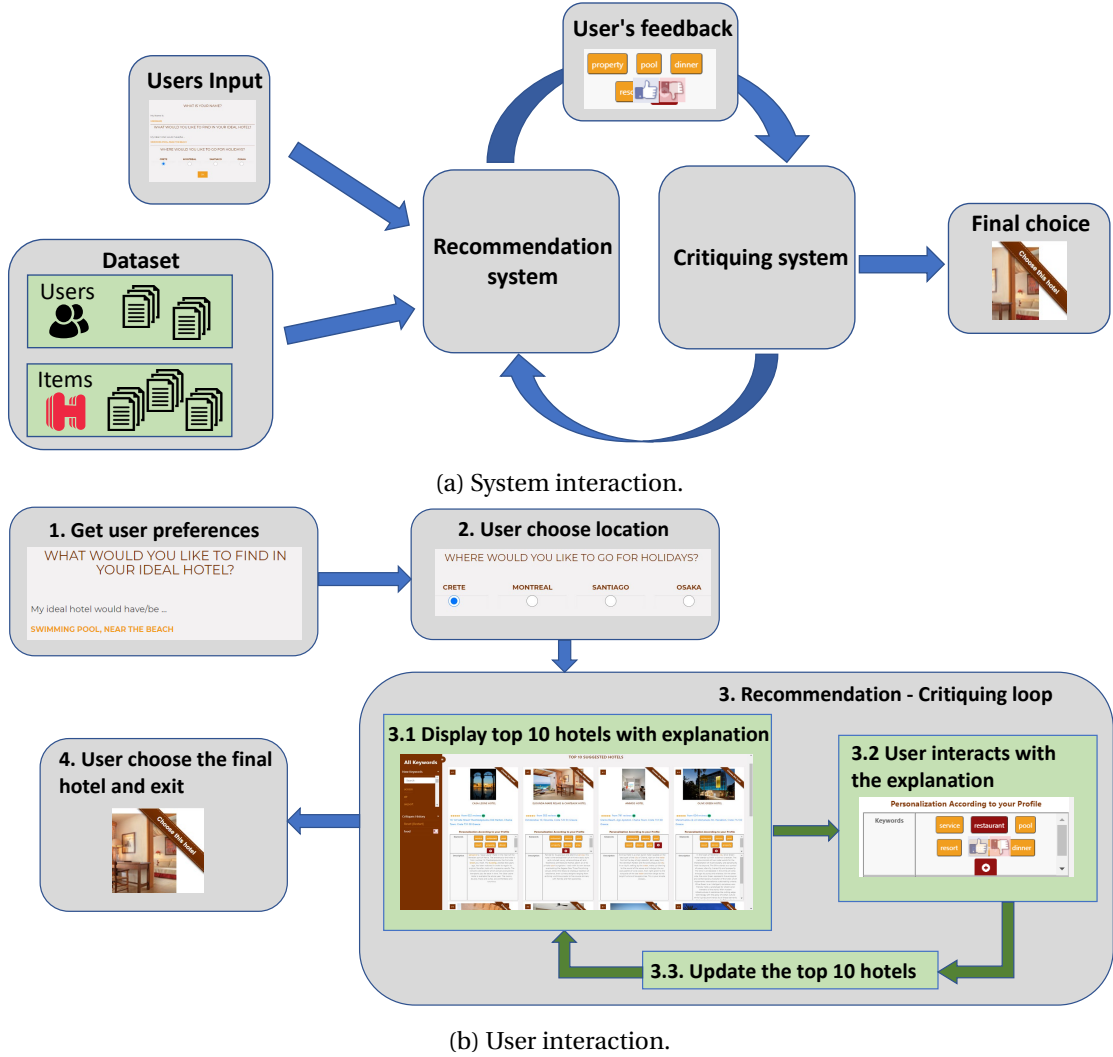


Figure 4.8: Interactions overview.

reviews from *TripAdvisor*, and automatically extracted 90 keyphrases to reflect hotel features (similar to those available on *TripAdvisor* and *Booking*).

Our system is model-agnostic both for the recommendation model and the critiquing mechanism. In case the recommender model does not handle the cold-start problem, we ask the users to provide an input to get their preferences and to define their initial profiles. We then match the query with an existing user in the database whose profile is most similar to the preferences formulated (according to a TF-IDF vector space model computed on user reviews). This enables more flexibility in the used recommender model.

We experiment with two recommender systems with critiquing mechanisms: T-RECS (Section 4.5) and CE-VAE (Luo et al., 2020b). The latter improves the training performance of CE-VNCF. Both predict a set of keywords from the 90 keyphrases extracted during preprocess-

ing, that best describes the user profile, alongside the recommended items. These keywords are therefore used as an explanation for the recommendation and can then be critiqued. However, these two models have their own specificities that we detail in Section 4.7.2.

The interaction with the interface from the user's point of view is shown in Figure 4.8b. Users are first asked to enter, in text form, what they would like for their ideal hotel. Then, users choose one of the four proposed destinations, where each contains approximately 25 to 45 hotels. Next, users enter the *Recommendation - Critiquing loop*. The interface initially displays only the top 10 items recommended based on initial user input, along with the generated explanation. The users have the possibility to interact with the explanations by critiquing them. After each critique, the system updates the recommendation of the users. This process continues until users are satisfied and cease to provide additional critiques. At this point, this iterative process terminates as we consider that users have found their ideal hotel.

### 4.7.2 Interface Details

Users can interact with four interfaces, detailed below and shown in Figure 4.9.

**Interface A: Static.** We use a static interface inspired by current hotel recommendation platforms as a reference point for user satisfaction. We allow users to filter hotels according to the keyword features (the same as the extracted ones during preprocessing). To make interaction easier, the keywords are grouped in 6 categories (see Figure 4.9a). The interface also displays the hotel's description scraped from *Tripadvisor*.

**Interface B: CE-VAE with shared keyphrases.** The recommender system predicts a set of keywords that best describe the user profile alongside the recommended items. For *CE-VAE*, users can only interact with the keywords on a global level to express an explicit disagreement: all items share the same explanation. The top 6 keywords associated with the user profile are displayed at the top of the web page (see Figure 4.9c). Similar to *Interface A*, the hotel descriptions are also displayed. However, the system's keywords are highlighted in the descriptions. It allows the users to also interact with these descriptions by critiquing them.

**Interface C: T-RECS with fine-grained keyphrases.** In contrast to *CE-VAE*, T-RECS generates more fine-grained explanations by inferring personalized keywords for each user-item pair (see Figure 4.9d). Moreover, users can critique a keyword to reflect a disagreement and highlight it as an important desired feature. As for *Interface B*, the hotel descriptions are also displayed and enable critiquing interactions.

**Interface D: T-RECS with fine-grained keyphrases and generated justifications.** This interface is similar to *Interface C* but, instead of displaying the general scrapped description for each hotel, it generates a personalized natural language justification conditioned on the keyword explanation. This justification, which is therefore based on user profiles, allows them to understand why those hotels are recommended. An example is shown in Figure 4.9e.

## 4.7 Multi-Step Critiquing User Interface

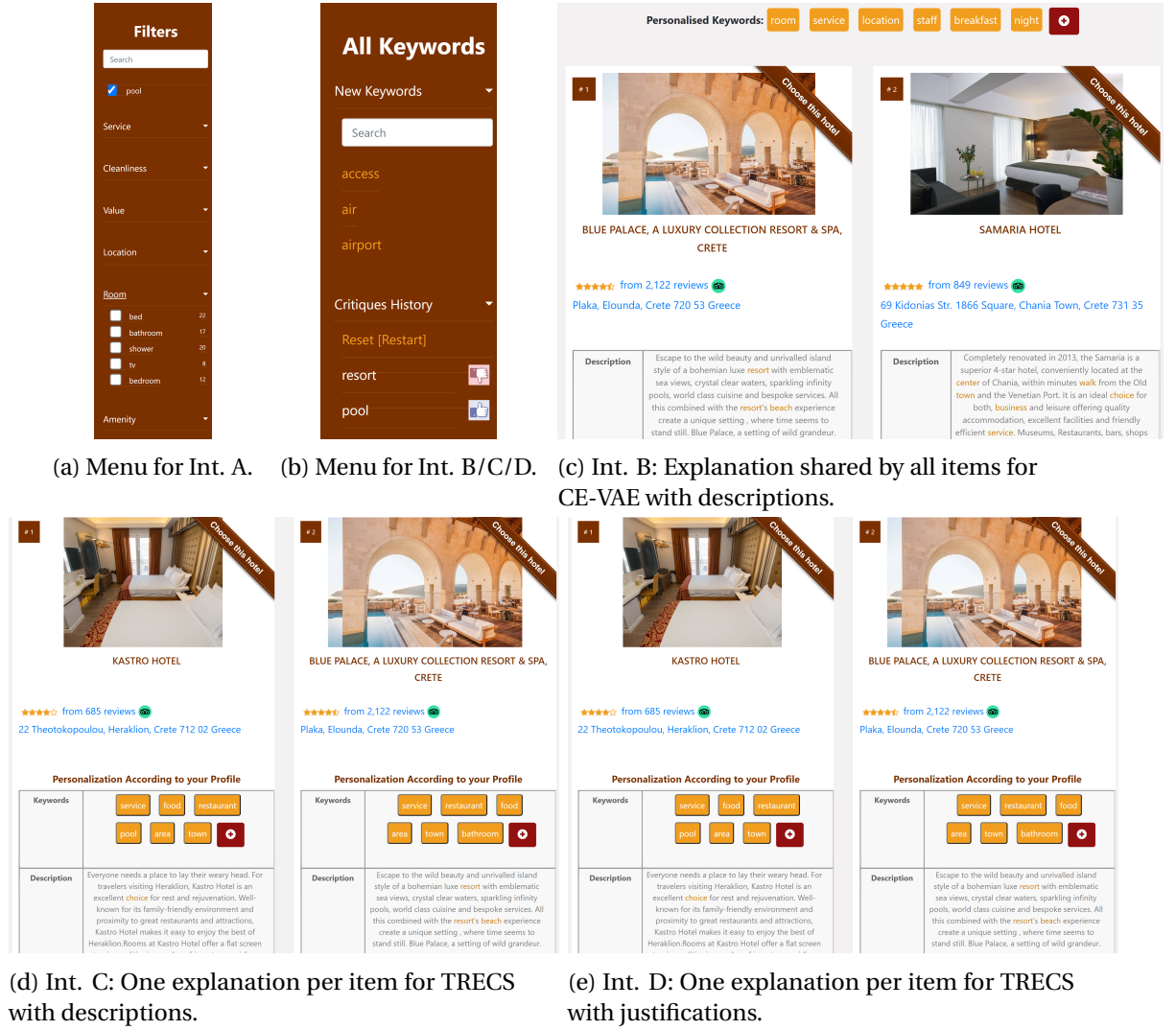


Figure 4.9: Interfaces.

In addition, for *Interfaces B, C, and D*, we give the possibility to search for additional keywords to critique (see Figure 4.9b) than those predicted as relevant by the system. Finally, after experimenting with the four interfaces, users access the entire hotel catalog to determine if the different approaches helped them find the hotel that best suits them.

### 4.7.3 Discussion

Some limitations of the natural language processing methods used can be observed by experimenting with the interfaces. We can take as an example the generated keywords. They are obtained by automated systems that generate them based on data mining techniques or machine learning models. However, meaningful keywords for the recommendation do not necessarily imply that they are helpful for the end user. On the one hand, the meanings of

individual keywords are not always clear for the end user, especially if they are not familiar with the recommendation domain. On the other hand, it is not always obvious what critiquing a keyword means. Indeed, the keyword “pool” indicates the presence or the absence of a pool. However, for the keyword “breakfast”, it is not clear how the system interprets breakfast as most hotels propose breakfast options. Post-processing to filter vague keywords or the usage of n-grams could improve the understanding of the keywords but partially solve the problem.

Moreover, with the recent progress of natural language generation, models are now capable of generating syntactically and grammatically correct sentences (Radford et al., 2019). However, we observe that the generated personalized justifications quite often lack diversity (Ni et al., 2019). Indeed, the justifications of the best items contain repeating parts of sentences. This makes it more difficult for the users to make a choice because the differences between the justifications are not emphasized. In addition, as is often the case in natural language applications, hallucinations can occur (Rohrbach et al., 2018; Holtzman et al., 2020).

### 4.8 Conclusion

Recommendations can carry much more impact if they are supported by explanations. In this chapter, we presented T-RECS, a multi-task learning Transformer-based recommender, that produces explanations considered significantly superior when evaluated by humans. The second contribution of T-RECS is the user’s ability to react to a recommendation by *critiquing* the explanation. We designed an unsupervised method for multi-step critiquing with explanations. Experiments show that T-RECS obtains stable and significant improvement in adapting to the preferences expressed in multi-step critiquing.

Finally, we designed several web interfaces that can be easily connected to different datasets, recommendation models, and critiquing mechanisms. In addition to the usual recommendations, they can provide users with keyword explanations and personalized natural language justifications. Users can interact with these keywords in different ways by critiquing them. It allows experimenting recommendations with critiquing based on neural natural language processing on a real use case. The interfaces are also a useful tool to highlight these systems’ limitations and find ways to overcome them.

# 5 Faster and Better Multi-Step Critiquing via Self-Supervision

## 5.1 Preface

**Contribution and Sources.** This chapter is largely based on Antognini and Faltings (2021a). The detailed individual contributions are listed below using the CRediT taxonomy (Brand et al., 2015) (terms are selected as applicable).

Diego Antognini (author): Conceptualization, Methodology, Software, Validation, Investigation, Formal Analysis, Writing – Original Draft, Writing – Review & Editing.

Boi Faltings: Writing – Review & Editing, Administration, Supervision.

**Summary.** Recent studies have shown that providing personalized explanations alongside recommendations increases trust and perceived quality. Furthermore, it gives users an opportunity to refine the recommendations by *critiquing* parts of the explanations. On one hand, current recommender systems model the recommendation, explanation, and critiquing objectives jointly, but this creates an inherent trade-off between their respective performance, and it yields poor results in multi-step critiquing settings. On the other hand, although recent latent linear critiquing approaches are built upon an existing recommender system, they suffer from computational inefficiency at inference due to the objective optimized at each conversation’s turn. In this chapter, we address these deficiencies with M&Ms-VAE, a novel variational autoencoder for recommendation and explanation that is based on multimodal modeling assumptions. We train the model under a weak supervision scheme to simulate both fully and partially observed variables. Then, we leverage the generalization ability of a trained M&Ms-VAE model to embed the user preference and the critique separately. Our work’s most important innovation is our critiquing module, which is built upon and trained in a self-supervised manner with a simple ranking objective. Experiments on four real-world datasets demonstrate that among state-of-the-art models, our system is the first to dominate or match the performance in terms of recommendation, explanation, and multi-step critiquing. Moreover, M&Ms-VAE processes the critiques up to 25.6x faster than

the best baselines. Finally, we show that our model infers coherent joint and cross generation, even under weak supervision, thanks to our multimodal-based modeling and training scheme.

### 5.2 Introduction

Recommender systems accurately capture user preferences and achieve high performance. However, they offer little transparency regarding their inner workings. It has been shown that providing explanations along with item recommendations enables users to understand why a particular item has been suggested and hence to make better decision (Chang et al., 2016; Bellini et al., 2018). Additionally, explanations increase the system’s overall transparency and trustworthiness (Tintarev and Masthoff, 2015; Zhang and Curley, 2018; Kunkel et al., 2018).

An important advantage of explanations is that they provide a basis for feedback. If users understand what has generated the suggestions, they can refine the recommendations by interacting directly with the explanations. Critiquing is a conversational recommendation method that incrementally adapts recommendations in response to user preferences (Chen and Pu, 2012). Example critiquing was introduced in information retrieval (Williams and Tou, 1982) and first applied to recommender systems in Burke et al. (1996). Recognizing that critiquing is most useful when applied in multiple steps, Linden et al. (1997) and Pu and Faltings (2000) introduced mechanisms based on constraint programming (Torrens et al., 2002) with an application to travel planning. Multi-step critiquing with constraint programming was recognized as a form of preference elicitation, which enabled the analysis and optimization of its performance (Faltings et al., 2004a) and the addition of suggestions for active preference elicitation (Viappiani et al., 2006), which yielded dramatic improvements in decision accuracy in user studies. Multi-step critiquing was also shown to be superior to compound critiquing, which groups multiple attributes in a single step (Reilly et al., 2007). A major limitation of all these approaches is that items have to be characterized by a set of discrete attributes.

After nearly a decade in which critiquing approaches received little attention, Wu et al. (2019a) introduced a collaborative filtering recommender with explanations and an embedding-based critiquing method. This method allows users to critique the recommendation using arbitrary languages; a set of attributes is mined from reviews, and the users can interact with them. Other works built upon the same paradigm (Antognini et al., 2021a; Chen et al., 2020). Luo et al. (2020b) showed that those models suffer from unstable training and high computational complexity, and they proposed a framework based on a variational autoencoder (Kingma and Welling, 2014; Liang et al., 2018). However, these models learn a bidirectional mapping between the critique and the user latent space. This creates an inherent trade-off between the recommendation and explanation performance, and it yields poor results in multi-step critiquing.

Recently, Luo et al. (2020a) proposed a latent linear critiquing (LLC) method built upon the recommendation model PLRec (Sedhain et al., 2016). LLC co-embeds keyphrase attributes in the same embedding space as the recommender. The critiquing process consists of a

weighted average between the user-preference embedding and the critique embeddings obtained through the conversation. The weights are optimized in a linear programming formulation using a max-margin scoring-based objective (i.e., the pairwise difference of scores of items affected by the critique and the others). Following the same methodology, Li et al. (2020) changed the objective into a ranking-based one. While those models obtain good performance in multi-step critiquing, they suffer from computational inefficiency due to the objective function optimized at each turn.

To address both issues, we present M&Ms-VAE, a novel variational autoencoder for recommendation and explanation with a separate critiquing module. Inspired by multimodal generative models (Wu and Goodman, 2018; Shi et al., 2019; Tsai et al., 2019; Sutter et al., 2020), we treat the user’s past interactions and keyphrase usage as different partially observed variables, and more importantly, we assume conditional independence between them. We can then approximate the variational joint posterior using a mixture of experts. We propose a training scheme that mimics weakly supervised learning to train the inference networks jointly but also independently. This is essential to our modeling, because M&Ms-VAE is robust to a missing unobserved variable and can thus embed separately and efficiently the user interactions, the keyphrases, and the critique, respectively.

In a second step, we leverage the generalization ability of M&Ms-VAE and design a novel blending module to re-rank recommended items according to a critique. The latter is trained only once on a synthetic dataset with a self-supervision objective. This generalizes into multi-step critiquing and enables fast critiquing.

To the best of our knowledge, this is the first work to revisit deep critiquing from the perspective of multimodal generative models and to propose a blending module trained in a simple self-supervised fashion. We evaluate our method using four real-world, publicly available datasets. The results demonstrate that the proposed M&Ms-VAE model

1. achieves superior or competitive performance in terms of recommendation, explanation, and multi-step critiquing in comparison to the state-of-the-art recommendation and critiquing methods,
2. processes the critiques up to 25.6x faster than the best baselines and up to 9.2x faster using only the CPU, and
3. induces coherent joint generation and cross generation, even under weak supervision.

### 5.3 Preliminaries

This section introduces the notation used in the chapter and the variational autoencoder for recommendation (Liang et al., 2018). Then, we review a recent study (Luo et al., 2020b) that built upon Liang et al. (2018) and revisited critiquing by proposing the critiquable-explainable VAE (CE-VAE) model. Finally, we highlight the key deficiencies that significantly limit its

performance in practice.

### 5.3.1 Notation

Before proceeding, we define the following notation used throughout this chapter:

- $U$ ,  $I$ , and  $K$ : The user, the item, and the keyphrase sets, respectively.
- $\mathbf{R} \in \mathbb{R}^{|U| \times |I|}$ : The user-by-item interaction matrix obtained with implicit feedback. Entries  $\mathbf{r}_{u,i}$  of 1 (respectively 0) denote a positive (respectively negative or unobserved) interaction between the user  $u$  and item  $i$ .
- $\mathbf{K} \in \mathbb{R}^{|U| \times |K|}$ : The binary user-keyphrase matrix that reflects the user  $u$ 's keyphrase-usage preference. Given user reviews from a corpus, we extract keyphrases that describe item attributes from all reviews (see Section 5.5.1).
- $\mathbf{K}^I \in \mathbb{R}^{|I| \times |K|}$ : The binary item-keyphrase matrix. The process is similar to  $\mathbf{K}$  with the aggregation per item.
- $\hat{\mathbf{r}}_u \in \mathbb{R}^{|I|}$  and  $\hat{\mathbf{k}}_u \in \mathbb{R}^{|K|}$ : The predicted feedback and keyphrase explanation, respectively.
- $\mathbf{z}_u \in \mathbb{R}^{|H|}$ : The user  $u$ 's latent embedding of dimension  $H$  from the observed interaction  $\mathbf{r}_u$  and keyphrase-usage preference  $\mathbf{k}_u$ .
- $\mathbf{z}_u^r \in \mathbb{R}^{|H|}$  and  $\mathbf{z}_u^k \in \mathbb{R}^{|H|}$ : The user  $u$ 's latent embedding of dimension  $H$  **only** from the observed interaction  $\mathbf{r}_u$  (symmetrically the keyphrase-usage preference  $\mathbf{k}_u$ ).
- $\mathbf{c}_u^t \in \mathbb{R}^{|K|}$ : A one-hot vector of length  $|K|$ . The only positive value indicates the index of the keyphrase to be critiqued by the user  $u$  at a given step  $t$  of the user interaction with the recommender system.
- $\mathbf{z}_u^t \in \mathbb{R}^{|H|}$ : The latent representation of the critique  $\mathbf{c}_u^t$ .
- $\tilde{\mathbf{z}}_u^t \in \mathbb{R}^{|H|}$ : The updated latent representation of the user after the critique  $\mathbf{c}_u^t$ .
- $I^{+\mathbf{c}} \in \{i | \mathbf{k}_{i,\mathbf{c}}^I = 1, \forall i \in I\}$ : The set of items that contain the critiqued keyphrase  $\mathbf{c}$ .
- $I^{-\mathbf{c}} \in \{i | \mathbf{k}_{i,\mathbf{c}}^I = 0, \forall i \in I\}$ : The set of items that do not contain the critiqued keyphrase  $\mathbf{c}$ .

### 5.3.2 Variational Autoencoder for Recommendation (VAE)

A variational autoencoder (VAE) (Kingma and Welling, 2014) is a generative model of the form  $p_\theta(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$ , where  $p(\mathbf{z})$  is a prior and the likelihood  $p_\theta(\mathbf{x}|\mathbf{z})$  is parametrized by a neural network with parameters  $\theta$ . The model learns to maximize the marginal likelihood of the data  $p_\theta(\mathbf{x})$  (i.e., the evidence that is intractable) by approximating the true unknown posterior  $p_\theta(\mathbf{z}|\mathbf{x})$  with a variational posterior  $q_\phi(\mathbf{z}|\mathbf{x})$ . Applied to recommendation systems,



the collaborative-filtering VAE (Liang et al., 2018) considers as input data the sparse user preferences  $\mathbf{r}_u$  over  $|I|$  items. More formally, the model optimizes a variational lower bound on the log likelihood of all observed user feedback  $\sum_{u \in U} \log p(\mathbf{r}_u)$  through stochastic gradient descent:

$$\begin{aligned} \log p(\mathbf{r}_u) &\geq \int_{\mathbf{z}_u} q_\phi(\mathbf{z}_u|\mathbf{r}_u) \log \frac{p_\theta(\mathbf{r}_u, \mathbf{z}_u)}{q_\phi(\mathbf{z}_u|\mathbf{r}_u)} d\mathbf{z}_u \\ &\geq \mathbb{E}_{q_\phi(\mathbf{z}_u|\mathbf{r}_u)} [\log p_\theta(\mathbf{r}_u|\mathbf{z}_u)] - \beta \text{D}_{\text{KL}}[q_\phi(\mathbf{z}_u|\mathbf{r}_u) \parallel p(\mathbf{z}_u)], \end{aligned} \quad (5.1)$$

where  $\mathbf{z}_u$  is sampled<sup>1</sup> from the distribution  $q_\phi(\mathbf{z}_u|\mathbf{r}_u)$  with parameters  $\mu_u$  and  $\Sigma_u$ , and  $\text{D}_{\text{KL}}[q, p]$  denotes the Kullback-Leibler divergence (KL) between the distributions  $p$  and  $q$ . In practice, the prior  $p(\mathbf{z})$  is usually a spherical Gaussian with parameters  $\mu$  and  $\Sigma$ . Finally,  $\beta$  is a hyperparameter that controls the strength of the regularization relative to the reconstruction error, as motivated by the  $\beta$ -VAE of Higgins et al. (2017), and is slowly annealed to 1, similarly to Bowman et al. (2016).

### 5.3.3 Co-embedding of Language-based Feedback with the Variational Autoencoder (CE-VAE)

Thus far, the variational autoencoder can only recommend items without generating any form of explanation. A recent study (Luo et al., 2020b) proposed the CE-VAE model, which integrates an explanation and critiquing module based on keyphrases. The authors support critiquing by first modeling the joint probability of a user’s item preferences and keyphrase usage:

$$\begin{aligned} \log p(\mathbf{r}_u, \mathbf{k}_u) &= \log p(\mathbf{k}_u|\mathbf{r}_u) + \log p(\mathbf{r}_u) \\ &= \mathbb{E}_{q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u)} [\log p_{\Theta_k}(\mathbf{k}_u|\mathbf{z}_u)] - \text{D}_{\text{KL}}[q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u) \parallel p(\mathbf{z}_u)] \\ &\quad + \mathbb{E}_{q_{\Psi_r}(\mathbf{z}_u|\mathbf{r}_u)} [\log p_{\Theta_r}(\mathbf{r}_u|\mathbf{z}_u)] + \mathcal{H}[q_{\Psi_r}(\mathbf{z}_u|\mathbf{r}_u)] + \mathbb{E}_{q_{\Psi_r}(\mathbf{z}_u|\mathbf{r}_u)} [p(\mathbf{z}_u)], \end{aligned} \quad (5.2)$$

where  $\mathcal{H}$  is the entropy. Then, they incorporate an additional objective to learn a projection from the critiquing feedback into the latent space via another encoder (an inverse feedback loop). In other words, they reintroduce the user’s keyphrase usage  $\mathbf{k}_u$  to approximate the variational lower bound of  $p(\mathbf{z}_u)$  by marginalizing over  $\mathbf{k}_u$ . More formally:

$$\begin{aligned} \log p(\mathbf{z}_u) &\geq \mathbb{E}_{q(\mathbf{k}_u|\mathbf{z}_u)} [\log p(\mathbf{z}_u|\mathbf{k}_u)] - \text{D}_{\text{KL}}[q(\mathbf{k}_u|\mathbf{z}_u) \parallel p(\mathbf{k}_u)] \\ &\approx \mathbb{E}_{p_{\Theta_k}(\mathbf{k}_u|\mathbf{z}_u)} [\log p_{\Theta'_k}(\mathbf{z}_u|\mathbf{k}_u)] - \text{D}_{\text{KL}}[p_{\Theta_k}(\mathbf{k}_u|\mathbf{z}_u) \parallel p(\mathbf{k}_u)], \end{aligned} \quad (5.3)$$

where  $p(\mathbf{k}_u)$  is a prior following a standard normal distribution and the weights of  $q(\mathbf{k}_u|\mathbf{z}_u)$  are shared with  $p_{\Theta_k}(\mathbf{k}_u|\mathbf{z}_u)$ .

Finally, once the model is trained on the full objective function, the critiquing process for the

<sup>1</sup>Using the reparametrization trick (Kingma and Welling, 2014; Rezende et al., 2014):  $\mathbf{z}_u = \mu_u + \epsilon \sigma_u$ , where  $\epsilon \sim \mathcal{N}(0, \mathbb{I}_H)$ .

critique  $\mathbf{c}_u$  is performed as follows:

1. compute the critique representation  $\mathbf{z}_u^c$  with  $p_{\Theta'_k}(\mathbf{z}_u|\mathbf{k}_u)$ ,
2. average both the user latent representation  $\mathbf{z}_u$  and the critique representation  $\mathbf{z}_u^c$ , and
3. predict the new feedback  $\hat{\mathbf{r}}_u$  with the generative network  $p_{\Theta_r}(\mathbf{r}_u|\mathbf{z}_u)$ .

Overall, the CE-VAE framework is effective in practice for recommendation, keyphrase explanation, and single-step critiquing. However, it suffers from two key deficiencies that limit its performance (as we later show empirically):

1. The model learns a function to project the critiqued keyphrase into the user's latent space, from which the feedback and the explanation are predicted. This mapping is learned via an autoencoder, which perturbs the training. Thus, there is an inherent trade-off between the performance of the recommendation and that of the explanation.
2. Although the joint objective also maximizes a latent representation likelihood with the Kullback-Leibler terms, it is unclear whether the inverse function embeds the critique effectively and whether the mean reflects a critiquing mechanism.<sup>2</sup>

### 5.4 M&Ms-VAE: A Mixture-of-Experts Multimodal Variational AutoEncoder

Our goal is to build a more generalizable representation of users' preferences that is based on their observed interactions and keyphrase usage. Figure 5.1 depicts the graphical model of our proposed M&Ms-VAE, and Figure 5.2 shows the training scheme. Then, we leverage this representation to efficiently embed the user critiques and learn, in a self-supervised fashion, a blending module to re-rank recommended items for multi-step critiquing. Figure 5.3 illustrates the workflow.

#### 5.4.1 Model Overview

Like previously developed variational autoencoders for recommendation, we assume that the observed user  $u$ 's interactions  $\mathbf{r}_u$  and the keyphrase-usage preference  $\mathbf{k}_u$  are generated from a latent representation of the user preferences.

Differently from prior work, we seek to learn the joint distribution  $p(\mathbf{r}_u, \mathbf{k}_u)$  under weak supervision. Our main goal is to learn a more generalizable representation of the user preferences. Therefore, we aim to design a generative model that can recommend and generate keyphrase

---

<sup>2</sup>The subsequent and concurrent work of Yang et al. (2021) has addressed these issues using Bayesian approximation. Although the recommender system is trained to predict ratings, it could be adapted for implicit feedback. For future work, we will adapt the model and include it in our experiments.

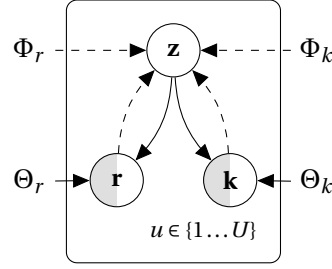


Figure 5.1: Probabilistic-graphical-model view of our M&Ms-VAE model. Both the implicit feedback  $\mathbf{r}_u$  and the keyphrase  $\mathbf{k}_u$  are generated from user  $u$ 's latent representation  $\mathbf{z}_u$ . Solid lines denote the generative model  $p_\theta$ , whereas dashed lines denote the variational approximation  $q_\phi$ .

explanation **jointly but also independently** from each of the observed variables (i.e., cross-modal generation). This condition is important as users are, most of the time, not required to write textual feedbacks of the consumed items. It also allows us to apply the same technique to users who have not written reviews or to cases in which keyphrases are unavailable. If this goal is achieved, we can then embed effectively the user's observed interactions, the user's keyphrase preference, and the critique with the **same** inference network  $q_\phi(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)$ .

Inspired by multimodal generative models (Wu and Goodman, 2018; Shi et al., 2019; Tsai et al., 2019; Sutter et al., 2020), we treat  $\mathbf{r}_u$  and  $\mathbf{k}_u$  as different modalities, and we assume they are conditionally independent given the common latent variable  $\mathbf{z}_u$ . In other words, we assume a generative model of the form  $p_\theta(\mathbf{r}_u, \mathbf{k}_u, \mathbf{z}_u) = p(\mathbf{z}_u) p_{\theta_r}(\mathbf{r}_u | \mathbf{z}_u) p_{\theta_k}(\mathbf{k}_u | \mathbf{z}_u)$ . An advantage of such a factorization is that if  $\mathbf{r}_u$  or  $\mathbf{k}_u$  is unobserved, we can safely ignore it when evaluating the marginal likelihood (Wu and Goodman, 2018).

We start with the derivation of the joint log likelihood  $\sum_{u \in U} \log p(\mathbf{r}_u, \mathbf{k}_u)$  over the observed interactions  $\mathbf{r}_u$  and keyphrase-usage preference  $\mathbf{k}_u$  and all users  $u$  as shown in Figure 5.1:

$$\begin{aligned} \log p(\mathbf{r}_u, \mathbf{k}_u) &= \log \int_{\mathbf{z}_u} p_\theta(\mathbf{r}_u, \mathbf{k}_u, \mathbf{z}_u) d\mathbf{z}_u \\ &\geq \mathbb{E}_{q_\phi(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} [\log p_\theta(\mathbf{r}_u, \mathbf{k}_u | \mathbf{z}_u)] - \beta \text{D}_{\text{KL}}[q_\phi(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u) \parallel p(\mathbf{z}_u)], \end{aligned} \quad (5.4)$$

where we assume that the prior distribution  $p(\mathbf{z})$  is a standard normal distribution and  $\beta$  is a hyperparameter that controls the strength of the regularization relative to the reconstruction error. Thanks to our assumption that  $\mathbf{r}_u$  and  $\mathbf{k}_u$  are conditionally independent given the common latent variable  $\mathbf{z}_u$ , we can rewrite Equation 5.4 as follows (see Appendix E for the complete derivation):

$$\begin{aligned} ELBO(\mathbf{r}_u, \mathbf{k}_u) &= \mathbb{E}_{q_\phi(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} [\log p_{\theta_r}(\mathbf{r}_u | \mathbf{z}_u) + \log p_{\theta_k}(\mathbf{k}_u | \mathbf{z}_u)] \\ &\quad - \beta \text{D}_{\text{KL}}[q_\phi(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u) \parallel p(\mathbf{z}_u)]. \end{aligned} \quad (5.5)$$

Learning the variational joint posterior  $q_\phi(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)$  of Equation 5.5 under its current form

requires  $\mathbf{r}_u$  and  $\mathbf{k}_u$  to be presented at all times, thus making cross-modal recommendation difficult. Following our assumption, we can factorize the joint variational posterior as a function  $\zeta(\cdot)$  of unimodal posteriors (or experts)  $q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u)$  and  $q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)$ :  $q_{\Phi}(\mathbf{z}_u|\mathbf{r}_u, \mathbf{k}_u) = \zeta(q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u), q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u))$ , similarly to Wu and Goodman (2018); Shi et al. (2019); Tsai et al. (2019); Sutter et al. (2020). In our case, the function  $\zeta(\cdot)$  should be 1. robust to overconfident experts if the marginal posterior  $q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u)$  or  $q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)$  has low density, and 2. robust to missing unobserved variable  $\mathbf{r}_u$  or  $\mathbf{k}_u$ . Therefore, we propose to rely on a mixture of experts (MoE) with uniform weights:

$$q_{\Phi}(\mathbf{z}_u|\mathbf{r}_u, \mathbf{k}_u) = \zeta(q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u), q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)) = \alpha \cdot q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u) + (1 - \alpha) \cdot q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)$$

$$\text{with } \alpha = \begin{cases} \frac{1}{2}, & \text{if } \mathbf{r}_u \text{ and } \mathbf{k}_u \text{ are observed,} \\ 1, & \text{if only } \mathbf{r}_u \text{ is observed,} \\ 0, & \text{if only } \mathbf{k}_u \text{ is observed.} \end{cases} \quad (5.6)$$

We set the weights uniformly to explicitly enforce an equal contribution from each  $\mathbf{r}_u$  and  $\mathbf{k}_u$  when both are observed during training. In the case of an unobserved modality, we shift the importance distribution toward the presented one, which generalizes to weakly supervised learning (see Section 5.4.2). This is an important factor, because the inference network  $q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)$  will later induce the critique representation. Finally, one might be tempted to learn  $\alpha$  jointly with the variational lower bound or dynamically. However, doing so might miscalibrate the precisions of the  $q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u)$  or  $q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)$  and thus be detrimental to the whole model in terms of both prediction performance and generalization.

### 5.4.2 Training Strategy

Combining Equations 5.5 and 5.6 gives the full objective function, and M&Ms-VAE can be trained on a complete dataset where all  $\mathbf{r}_u$  and  $\mathbf{k}_u$  are provided. However, in doing so, we never train the individual inference networks  $q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u)$  and  $q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)$ ; only the relationship between the observed user interactions and keyphrase-usage preferences is captured. As a consequence, at inference, it is unclear how the model performs with a missing observation.

To reach our goal of recommending given at least  $\mathbf{r}_u$  and embedding the critique effectively with the inference network  $q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)$ , we propose a training strategy that mimics weakly supervised learning, similarly to Wu and Goodman (2018). Moreover, this allows us to handle incomplete datasets, where some samples are partially observed: data that contain only  $\mathbf{r}_u$  or  $\mathbf{k}_u$ <sup>3</sup>. The training strategy is shown in Figure 5.2. For each minibatch, we compute the gradient on the evidence lower bound of the joint observation and each single observation  $\mathbf{r}_u$  and  $\mathbf{k}_u$ .

---

<sup>3</sup>It also enables another way to solve the cold-start problem: new users can select a set of items and/or relevant keyphrases that reflect their preferences.

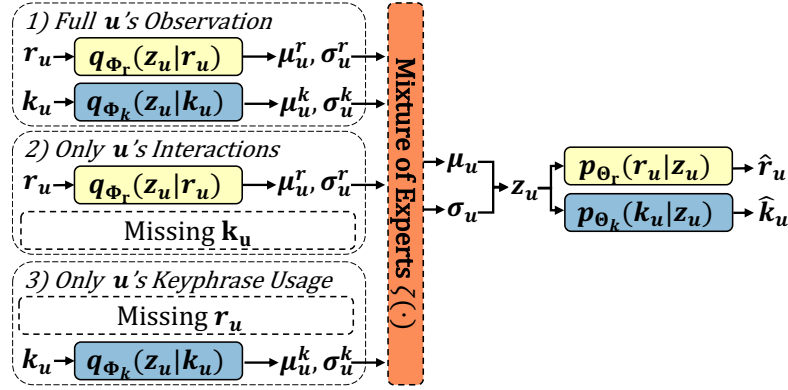


Figure 5.2: The proposed M&Ms-VAE architecture and training scheme. Each pass infers the parameters  $\mu_u$  and  $\sigma_u$  with the mixture of experts using either the joint inference network  $q_\Phi(\mathbf{z}_u|\mathbf{r}_u, \mathbf{k}_u)$  or one of the individual networks ( $q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u)$  or  $q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)$ , respectively). The final gradient is computed on the sum of each  $ELBO(\cdot)$  term.

Our final training objective for all users  $u$  is

$$\begin{aligned}
 \mathcal{L}(\mathbf{R}, \mathbf{K}) = & \sum_{u \in U} \underbrace{\lambda \cdot \mathbb{E}_{q_\Phi(\mathbf{z}_u|\mathbf{r}_u, \mathbf{k}_u)} [\log p_{\Theta_r}(\mathbf{r}_u|\mathbf{z}_u) + \log p_{\Theta_k}(\mathbf{k}_u|\mathbf{z}_u)] - \beta \text{D}_{\text{KL}}[q_\Phi(\mathbf{z}_u|\mathbf{r}_u, \mathbf{k}_u) \parallel p(\mathbf{z}_u)]}_{ELBO(\mathbf{r}_u, \mathbf{k}_u)} \\
 & + \sum_{u \in U} \underbrace{\lambda \cdot \mathbb{E}_{q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u)} [\log p_{\Theta_r}(\mathbf{r}_u|\mathbf{z}_u)] - \beta \text{D}_{\text{KL}}[q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u) \parallel p(\mathbf{z}_u)]}_{ELBO(\mathbf{r}_u)} \\
 & + \sum_{u \in U} \underbrace{\lambda \cdot \mathbb{E}_{q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)} [\log p_{\Theta_k}(\mathbf{k}_u|\mathbf{z}_u)] - \beta \text{D}_{\text{KL}}[q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u) \parallel p(\mathbf{z}_u)]}_{ELBO(\mathbf{k}_u)},
 \end{aligned} \tag{5.7}$$

where  $\lambda$  and  $\beta$  control the strength of the reconstruction error and regularization, respectively.

#### 5.4.3 Self-Supervised Critiquing with M&Ms-VAE

The purpose of critiquing is to refine the recommendation  $\hat{\mathbf{r}}_u$  based on the user  $u$ 's interaction with the explanation  $\hat{\mathbf{k}}_u$ , represented with a binary vector. The user can accept the recommended items, at which point the session terminates. In the other case, the user can provide a critique  $\mathbf{c}_u^t$  and obtain a new recommendation  $\hat{\mathbf{r}}_u^t$ . The process is repeated over  $T$  iterations until the user  $u$  is satisfied with the recommendation. Each critique  $\mathbf{c}_u^t$  is encoded as a one-hot vector where the positive value indicates a keyphrase the user  $u$  dislikes. The overall process is depicted in Figure 5.3.

We leverage the generalization ability of the trained M&Ms-VAE, especially the inference models  $q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u)$  and  $q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)$ . We use the former to represent the initial user preferences  $\mathbf{r}_u$  and the latter to embed the critique  $\mathbf{c}_u^t$ . However, a crucial question remains: how should

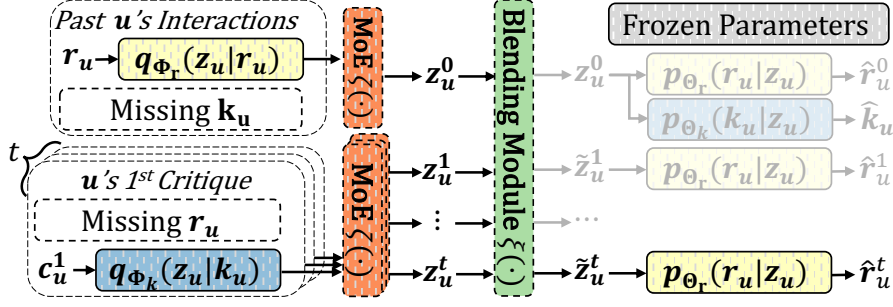


Figure 5.3: Workflow of considering the recommendation of items to a user  $u$  over  $t+1$  time steps. First, M&Ms-VAE produces the initial set of recommended items  $\hat{\mathbf{r}}_u^0 = \hat{\mathbf{r}}_u$  using only the historical observed interactions  $\mathbf{r}_u$ . Then, the user can provide a critique  $\mathbf{c}_u^t$  that is encoded into  $\mathbf{z}_u^t$  via the inference model  $q_{\Phi_k}(\mathbf{z}_u | \mathbf{k}_u)$ . The blending module combines the previous representations  $\mathbf{z}_u^0, \mathbf{z}_u^1, \dots, \mathbf{z}_u^t$  into  $\tilde{\mathbf{z}}_u^t$ , from which the subsequent recommendation  $\hat{\mathbf{r}}_u^t$  is computed. This process continues until the user  $u$  accepts the recommendation and ceases to provide additional critiques.

we blend the user representation  $\mathbf{z}_u$  with the  $t^{\text{th}}$  critique representation  $\mathbf{z}_u^t$ ?

Prior work has implemented a blending function as a simple average (Wu et al., 2019a; Luo et al., 2020b) or as a linear programming task that looks for a convex combination of embeddings provided with a specific linear optimization objective (Luo et al., 2020a; Li et al., 2020). As we demonstrate empirically later, the former yields poor performance when iterated for multi-step critiquing, whereas the latter is computationally slow because the optimization is performed for each critique and it cannot leverage GPUs.

### Blending Function Design.

We consider the critiquing process as a short session. We propose to learn a blending function  $\xi(\cdot)$  built upon a trained M&Ms-VAE model whose weights are frozen. This two-step approach has several advantages:

1. The original training of M&Ms-VAE is not perturbed by the critiquing objective.
2.  $\xi(\cdot)$  is decoupled from the model. It allows more flexibility in its architecture, objective function, and training.

We assume that each critique is independent, as in Luo et al. (2020a); Wu et al. (2019a); Antognini et al. (2021a). At time step  $t$ , we express the new user preferences  $\tilde{\mathbf{z}}_u^t$  with a linear interpolation between the original latent representation  $\mathbf{z}_u^0$ , inferred from the observed interaction  $\mathbf{r}_u$  **only**, and the critique  $\mathbf{c}_u^t$ 's representation  $\mathbf{z}_u^t$ . More precisely, we use the gating mechanism of the gated recurrent unit (Chung et al., 2014) (we omit the biases to reduce

---

**Algorithm 2** Synthetic Critiquing Dataset Creation
 

---

```

1: function GENERATE( $\mathbf{R}^{\text{val}}, \mathbf{K}^I$ )
2:   Synthetic dataset  $D \leftarrow \{\}$ 
3:   for each user  $u$  do
4:     for each target item  $i$ , where  $\mathbf{r}_{u,i}^{\text{val}} = 1$  do
5:       Randomly sample a critique  $\mathbf{c} \in K \setminus \mathbf{k}_i^I$ 
6:       Compute the item sets  $I^{+\mathbf{c}}$  and  $I^{-\mathbf{c}}$ 
7:       Update  $D \leftarrow D \cup \{(u, i, \mathbf{c}, I^{+\mathbf{c}}, I^{-\mathbf{c}})\}$ 
8:   return Synthetic dataset  $D$ 
    
```

---

notational clutter):

$$\begin{aligned}
 \tilde{\mathbf{z}}_u^t = \xi(\mathbf{z}_u^0, \mathbf{z}_u^t) = h_2 \quad & \begin{aligned} h_2 &= (1 - u_1) \cdot n_1 + u_1 \cdot h_1 \\ n_1 &= \tanh(W_{in}\mathbf{z}_u^t + W_{hn}(r_1 \odot h_1)) \\ u_1 &= \sigma(W_{iu}\mathbf{z}_u^t + W_{hz}h_1) \\ r_1 &= \sigma(W_{ir}\mathbf{z}_u^t + W_{hr}h_1) \end{aligned} \quad & \begin{aligned} h_1 &= (1 - u_0) \cdot n_0 + u_0 \cdot h_0 \\ n_0 &= \tanh(W_{in}\mathbf{z}_u^0 + W_{hn}(r_0 \odot h_0)) \\ u_0 &= \sigma(W_{iu}\mathbf{z}_u^0 + W_{hz}h_0) \\ r_0 &= \sigma(W_{ir}\mathbf{z}_u^0 + W_{hr}h_0) \end{aligned} \quad , \\
 \end{aligned} \tag{5.8}$$

where  $W_{ir}, W_{iu}, W_{in}, W_{hr}, W_{hz}, W_{hn}$ , and the bias vectors are the model parameters.

### Training.

Thanks to our assumption and the generalization ability of M&Ms-VAE, we can learn the weights of the blending module  $\xi(\cdot)$  by creating a synthetic dataset based only on the validation set (see Algorithm 2). For each user and observed interaction, we randomly sample a keyphrase  $\mathbf{c}$  that is inconsistent with the target item. Then, we calculate the item sets  $I^{+\mathbf{c}}$  that contain the critique and, symmetrically, the item sets  $I^{-\mathbf{c}}$  for those that do not contain it<sup>4</sup>.

Our final objective is to re-rank items based on the user preferences and the provided critique  $\mathbf{c}$ . Recall that M&Ms-VAE's weights are frozen. Let  $\hat{\mathbf{r}}_u^0$  be the user  $u$ 's initial predictions  $\hat{\mathbf{r}}_u$  and  $\hat{\mathbf{r}}_u^1$  those inferred from  $\tilde{\mathbf{z}}_u^t$  after the critique. We express this overall ranking-based objective via two differentiable max-margin objective functions (Rosasco et al., 2004). We train  $\xi(\cdot)$  and minimize the following loss function:

$$\begin{aligned}
 \mathcal{L}(\hat{\mathbf{R}}^0, \hat{\mathbf{R}}^1, u, \mathbf{c}, I^{+\mathbf{c}}, I^{-\mathbf{c}}) &= \mathcal{L}(p_{\Theta_r}(\mathbf{z}_u^0), p_{\Theta_r}(\xi(\mathbf{z}_u^0, \mathbf{z}_u^t)), u, \mathbf{c}, I^{+\mathbf{c}}, I^{-\mathbf{c}}) \\
 &= \sum_{i^+ \in I^{+\mathbf{c}}} \left[ \max \left\{ 0, h - (\hat{\mathbf{r}}_{u,i^+}^0 - \hat{\mathbf{r}}_{u,i^+}^1) \right\} \right] \\
 &\quad + \sum_{i^- \in I^{-\mathbf{c}}} \left[ \max \left\{ 0, h - (\hat{\mathbf{r}}_{u,i^-}^1 - \hat{\mathbf{r}}_{u,i^-}^0) \right\} \right], \tag{5.9}
 \end{aligned}$$

---

<sup>4</sup>In early experiments, we generalized  $\xi(\mathbf{z}_u^0, \mathbf{z}_u^t)$  to  $\xi(\mathbf{z}_u^0, \mathbf{z}_u^1, \dots, \mathbf{z}_u^t)$  and updated Algorithm 2 accordingly. However, our synthetic dataset cannot cover such a space due to the exponential number of combinations; session-based recommenders require millions of real sessions as training data (Hidasi et al., 2016; Hidasi and Karatzoglou, 2018).

Table 5.1: Descriptive statistics of the datasets. Coverage shows the ratio of reviews having at least one of the selected keyphrases (KPs)

Dataset	#Users	#Items	#Interactions	Sparsity	#KPs	KP Coverage	Avg. KPs/Review	AVG. KPs/User
Beer	6,370	3,669	263,244	1.13%	75	99.27%	7.16	1,216
CDs&Vinyl	6,060	4,395	152,783	0.57%	40	74.59%	2.13	73
Yelp	9,801	4,706	140,496	0.30%	234	96.65%	7.45	300
Hotel	7,044	4,874	143,612	0.42%	141	99.99%	17.42	419

where  $h$  is the margin. Intuitively,  $\xi(\cdot)$  is encouraged to create a representation  $\tilde{\mathbf{z}}_u^t$  from which  $p_{\Theta_r}(\cdot)$  gives a lower ranking to the items affected by the critique in the next iteration (i.e.,  $\hat{\mathbf{r}}_{u,i^+}^1 < \hat{\mathbf{r}}_{u,i^+}^0$ ) and a higher ranking to the unaffected items (i.e.,  $\hat{\mathbf{r}}_{u,i^-}^1 > \hat{\mathbf{r}}_{u,i^-}^0$ ). Finally, Equation 5.9 is efficiently parallelizable on both CPUs and GPUs.

## 5.5 Experiments

In this section, we proceed to evaluate the proposed M&Ms-VAE model in order to answer the following questions:

- **RQ 1:** How does M&Ms-VAE perform in terms of recommendation and explanation performance?
- **RQ 2:** Can M&Ms-VAE with the self-supervised critiquing objective enable multi-step critiquing?
- **RQ 3:** What is our proposed critiquing algorithm’s computational time complexity compared to prior work?
- **RQ 4:** How does M&Ms-VAE perform under weak supervision; how coherent is the joint and cross generation?

### 5.5.1 Datasets

We evaluate the quantitative performance of M&Ms-VAE using four real-world, publicly available datasets: BeerAdvocate (McAuley et al., 2012), Amazon CDs&Vinyl (McAuley et al., 2015; He and McAuley, 2016), Yelp (Yelp, 2016), and HotelRec (Antognini and Faltings, 2020b). Each contains more than 100k reviews with five-star ratings. For the purpose of Top-N recommendation, we binarize the ratings with a threshold  $t > 3.5$ . Because people tend to rate beers and restaurants positively, we set the threshold  $t > 4$  and  $t > 4.5$ , respectively. We split each dataset into 60%/20%/20% for the training, validation, and test sets. Table 5.1 shows the data statistics. All contain complete observations. The datasets do not contain preselected keyphrases. Hence, we extract them for the explanations and critiquing with the frequency-based processing of Wu et al. (2019a); Li et al. (2020). Some examples are shown in Table 5.2.



Table 5.2: Some keyphrases mined from the reviews. We manually grouped them by types for a better understanding.

Dataset	Type	Keyphrases
Beer	Head	white, tan, offwhite, brown
	Malt	roasted, caramel, pale, wheat, rye
	Color	golden, copper, orange, black, yellow
	Taste	citrus, fruit, chocolate, cherry, plum
CDs&Vinyl	Genre	rock, pop, jazz, rap, hip hop, R&B
	Instrument	orchestra, drum
	Style	concert, opera
	Religious	chorus, christian, gospel
Yelp	Cuisine	chinese, thai, italian, mexican, french
	Drink	tea, coffee, bubble tea, wine, soft drinks
	Food	chicken, beef, fish, pork, seafood, cheese
	Price & Service	cheap, pricy, expensive, busy, friendly
Hotel	Service	bar, lobby, housekeeping, guest, shuttle
	Cleanliness	toilet, sink, tub, smoking, toiletry, bathroom
	Location	airport, downtown, city, shop, restaurant
	Room	bed, tv, balcony, terrace, kitchen, business

### 5.5.2 Experimental Settings

Across experiments, we treat the prior and the likelihood as standard normal and multinomial distributions, respectively. The inference and generative networks consist of a two-layer neural network with a tanh nonlinearity as the activation function between the layers. We normalize the input and use dropout (Srivastava et al., 2014). For learning, we employ the Adam optimizer (Kingma and Ba, 2015) with AMSGrad (Reddi et al., 2018) and a learning rate of  $5 \cdot 10^{-5}$ . We anneal linearly the regularization parameter  $\beta$  of the Kullback-Leibler terms. For the baselines, we reused the authors' code and tuning procedure. We select hyperparameters and architectures for each model by evaluating NDCG on the validation set. We limit the search to a maximum of 100 trials. For critiquing, we tune our blending module on the synthetic dataset with the Falling MAP metric on the validation set, which measures the effect of a critique (Wu et al., 2019a). For reproducibility purposes, we include additional details and the best hyperparameters in Appendix F.4.

### 5.5.3 RQ 1: How does M&Ms-VAE perform in terms of recommendation and explanation performance?

In this section, we investigate whether the factorization and the training strategy of our proposed M&Ms-VAE model benefit the recommendation and explanation performance.

### Baselines.

We compare our proposed M&Ms-VAE model to the following baseline models. **POP** returns the most popular items without any kind of personalization. **AutoRec** (Sedhain et al., 2015) is a neural autoencoder-based recommendation system. **BPR** (Rendle et al., 2009) is a Bayesian personalized ranking model that explicitly optimizes pairwise rankings. **CDAE** (Wu et al., 2016) denotes a collaborative denoising autoencoder that is specifically optimized for implicit feedback recommendation tasks. **NCE-PLRec** (Wu et al., 2019b) represents the linear recommendation projected by noise-contrastive estimation; it augments PLRec with noise-contrasted item embeddings. **PLRec** (Sedhain et al., 2016) is the ablation variant of NCE-PLRec without the noise-contrastive estimation. **PureSVD** (Cremonesi et al., 2010) denotes a similarity-based recommendation method that constructs a similarity matrix through SVD decomposition of the implicit rating matrix. **VAE-CF** is the variational autoencoder for collaborative filtering described in Section 5.3.2. **CE-VNCF** (Wu et al., 2019a) is the extension of the neural collaborative filtering model (He et al., 2017) that is augmented with an explanation and a critiquing neural component. Finally, **CE-VAE** (Luo et al., 2020b) is a significant improvement over CE-VNCF, and it produces state-of-the-art performance (more details in Section 5.3.3). For a fair comparison, we encode the user observations in M&Ms-VAE using solely the inference network  $q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u)$  at test time. We provide in Section 5.5.6 the results with the other settings.

### Top-N Recommendation Performance.

We report the following metrics: R-Precision and NDCG; MAP, Precision, and Recall at different Top-N. The main results are presented in Table 5.3. We make the following key observations.

Overall, M&Ms-VAE shows the best recommendation performance for all metrics on three datasets and nearly all metrics on the CDs&Vinyl dataset. Compared to the original VAE recommender (VAE-CF), M&Ms-VAE achieves an improvement of 13% on average. We conjecture that the extra loss terms ( $ELBO(\mathbf{r}_u, \mathbf{k}_u)$  and  $ELBO(\mathbf{k}_u)$ ) help to generate better user representations by leveraging both user preferences and keyphrase usage with the mixture of experts.

M&Ms-VAE also significantly outperforms CE-VAE on the Yelp and Hotel datasets (by a factor of 1.9 and 1.7, respectively) and achieves an average improvement of 9% on the Beer and CDs&Vinyl datasets. We remark the same trend with CE-VNCF. These results emphasize the noise introduced in CE-VAE and CE-VNCF during training when learning the mapping between the keyphrases and the latent space. This is even more pronounced with a large number of keyphrases (i.e., over 100). In contrast, M&Ms-VAE is more robust thanks to our factorization and training strategy, which mimics a weak supervision setting;

Interestingly, PureSVD exhibits the second-best performance on the CDs&Vinyl and Yelp datasets. This shows that classic algorithms often remain competitive with state-of-the-art VAE-based recommender systems.

Table 5.3: Top-N recommendation results of all datasets. **Bold** and underline denote the best and second-best results, respectively. We omit the error bars because the 95% confidence interval is in 4<sup>th</sup> digit.

	Model	R-Precision	NDCG	MAP@N			Precision@N			Recall@N		
				N = 5	N = 10	N = 20	N = 5	N = 10	N = 20	N = 5	N = 10	N = 20
<i>Beer</i>	POP	0.0307	0.0777	0.0388	0.0350	0.0319	0.0346	0.0298	0.0279	0.0241	0.0408	0.0737
	AutoRec	0.0496	0.1140	0.0652	0.0591	0.0527	0.0574	0.0503	0.0438	0.0392	0.0663	0.1129
	BPR	0.0520	0.1214	0.0646	0.0597	0.0538	0.0596	0.0525	0.0449	0.0451	0.0744	0.1214
	CDAE	0.0414	0.0982	0.0504	0.0477	0.0434	0.0482	0.0432	0.0368	0.0330	0.0576	0.0969
	NCE-PLRec	0.0501	0.1151	0.0643	0.0594	0.0532	0.0589	0.0518	0.0440	0.0418	0.0714	0.1177
	PLRec	0.0497	0.1113	0.0655	0.0599	0.0532	0.0590	0.0515	0.0431	0.0421	0.0704	0.1127
	PureSVD	0.0450	0.1052	0.0479	0.0473	0.0446	0.0493	0.0455	0.0396	0.0391	0.0689	0.1131
	VAE-CF	<u>0.0538</u>	<u>0.1275</u>	0.0642	0.0594	0.0536	0.0595	0.0525	0.0448	<u>0.0473</u>	<u>0.0808</u>	<u>0.1327</u>
	CE-VAE	0.0520	0.1215	<u>0.0675</u>	<u>0.0618</u>	<u>0.0555</u>	<u>0.0620</u>	<u>0.0536</u>	<u>0.0461</u>	<u>0.0442</u>	<u>0.0737</u>	<u>0.1255</u>
	CE-VNCF	0.0440	0.1099	0.0546	0.0512	0.0472	0.0504	0.0465	0.0411	0.0353	0.0635	0.1116
	M&Ms-VAE (Ours)	<b>0.0545</b>	<b>0.1307</b>	<b>0.0706</b>	<b>0.0650</b>	<b>0.0580</b>	<b>0.0649</b>	<b>0.0563</b>	<b>0.0473</b>	<b>0.0492</b>	<b>0.0833</b>	<b>0.1349</b>
<i>CDs&amp;Vinyl</i>	POP	0.0088	0.0265	0.0108	0.0102	0.0095	0.0098	0.0095	0.0082	0.0088	0.0182	0.0327
	AutoRec	0.0227	0.0537	0.0284	0.0257	0.0220	0.0255	0.0213	0.0165	0.0254	0.0418	0.0627
	BPR	0.0632	0.1516	0.0724	0.0639	0.0543	0.0640	0.0513	0.0408	0.0807	0.1263	0.1939
	CDAE	0.0135	0.0365	0.0173	0.0158	0.0141	0.0152	0.0136	0.0116	0.0143	0.0262	0.0451
	NCE-PLRec	0.0749	<u>0.1739</u>	0.0728	0.0678	0.0586	0.0698	0.0584	0.0441	<b>0.1010</b>	<b>0.1608</b>	<u>0.2308</u>
	PLRec	<u>0.0760</u>	0.1626	<b>0.0889</b>	<u>0.0777</u>	<u>0.0642</u>	<u>0.0773</u>	<u>0.0608</u>	<u>0.0444</u>	0.0960	0.1461	0.2025
	PureSVD	0.0652	0.1551	0.0570	0.0565	0.0509	0.0612	0.0527	0.0405	0.0914	0.1486	0.2149
	VAE-CF	0.0638	0.1699	0.0540	0.0554	0.0517	0.0600	0.0540	0.0440	0.0949	<u>0.1593</u>	<b>0.2381</b>
	CE-VAE	0.0708	0.1532	0.0816	0.0711	0.0588	0.0715	0.0555	0.0411	0.0903	0.1357	0.1937
	CE-VNCF	0.0654	0.1524	0.0746	0.0662	0.0560	0.0663	0.0534	0.0411	0.0829	0.1299	0.1931
	M&Ms-VAE (Ours)	<b>0.0801</b>	<b>0.1765</b>	<u>0.0885</u>	<b>0.0784</b>	<b>0.0660</b>	<b>0.0779</b>	<b>0.0628</b>	<b>0.0482</b>	<u>0.0983</u>	0.1529	0.2263
<i>Yelp</i>	POP	0.0026	0.0129	0.0024	0.0026	0.0026	0.0028	0.0028	0.0025	0.0042	0.0087	0.0151
	AutoRec	0.0034	0.0133	0.0032	0.0030	0.0028	0.0027	0.0027	0.0025	0.0038	0.0081	0.0153
	BPR	0.0160	0.0609	0.0168	0.0156	0.0143	0.0156	0.0140	0.0122	0.0236	0.0435	0.0748
	CDAE	0.0028	0.0135	0.0027	0.0028	0.0027	0.0030	0.0027	0.0026	0.0044	0.0084	0.0161
	NCE-PLRec	0.0197	0.0739	0.0220	0.0200	0.0177	0.0198	0.0169	0.0143	0.0300	0.0505	0.0864
	PLRec	0.0191	0.0703	0.0207	0.0189	0.0171	0.0185	0.0166	0.0143	0.0291	0.0513	0.0866
	PureSVD	<u>0.0253</u>	<u>0.0825</u>	<u>0.0279</u>	<u>0.0249</u>	<u>0.0217</u>	<u>0.0240</u>	<u>0.0206</u>	<u>0.0173</u>	<u>0.0357</u>	<u>0.0597</u>	<u>0.1008</u>
	VAE-CF	0.0214	0.0801	0.0232	0.0216	0.0195	0.0214	0.0192	0.0163	0.0319	0.0589	0.0995
	CE-VAE	0.0136	0.0533	0.0143	0.0132	0.0121	0.0125	0.0119	0.0104	0.0197	0.0367	0.0636
	CE-VNCF	0.0166	0.0693	0.0175	0.0167	0.0157	0.0165	0.0154	0.0144	0.0251	0.0467	0.0889
	M&Ms-VAE (Ours)	<b>0.0264</b>	<b>0.0909</b>	<b>0.0284</b>	<b>0.0261</b>	<b>0.0231</b>	<b>0.0260</b>	<b>0.0223</b>	<b>0.0188</b>	<b>0.0395</b>	<b>0.0682</b>	<b>0.1154</b>
<i>Hotel</i>	POP	0.0047	0.0188	0.0049	0.0047	0.0042	0.0047	0.0043	0.0036	0.0054	0.0098	0.0167
	AutoRec	0.0051	0.0193	0.0053	0.0050	0.0044	0.0052	0.0042	0.0037	0.0061	0.0097	0.0169
	BPR	0.0181	0.0623	0.0198	0.0185	0.0169	0.0183	0.0168	0.0146	0.0219	0.0409	0.0713
	CDAE	0.0050	0.0190	0.0054	0.0049	0.0044	0.0050	0.0043	0.0037	0.0057	0.0098	0.0172
	NCE-PLRec	0.0229	0.0684	0.0244	0.0226	0.0200	0.0228	0.0195	0.0160	0.0283	0.0484	0.0785
	PLRec	0.0242	0.0664	0.0265	0.0234	0.0201	0.0228	0.0190	0.0155	0.0284	0.0466	0.0758
	PureSVD	0.0179	0.0541	0.0193	0.0173	0.0152	0.0169	0.0145	0.0121	0.0206	0.0357	0.0594
	VAE-CF	<u>0.0243</u>	<u>0.0755</u>	<u>0.0267</u>	<u>0.0241</u>	<u>0.0213</u>	<u>0.0238</u>	<u>0.0206</u>	<u>0.0171</u>	<u>0.0295</u>	<u>0.0511</u>	<u>0.0848</u>
	CE-VAE	0.0147	0.0538	0.0151	0.0146	0.0136	0.0148	0.0137	0.0122	0.0184	0.0334	0.0595
	CE-VNCF	0.0165	0.0575	0.0180	0.0166	0.0152	0.0159	0.0149	0.0129	0.0200	0.0370	0.0635
	M&Ms-VAE (Ours)	<b>0.0272</b>	<b>0.0804</b>	<b>0.0290</b>	<b>0.0265</b>	<b>0.0235</b>	<b>0.0260</b>	<b>0.0227</b>	<b>0.0189</b>	<b>0.0314</b>	<b>0.0555</b>	<b>0.0928</b>

Table 5.4: Top-K keyphrase explanation quality of all datasets. **Bold** and underline denote the best and second-best results, respectively. We omit the error bars because the 95% confidence interval is in 4<sup>th</sup> digit.

		NDCG@K			MAP@K			Precision@K			Recall@K		
Model		K = 5	K = 10	K = 20	K = 5	K = 10	K = 20	K = 5	K = 10	K = 20	K = 5	K = 10	K = 20
Beer	UserPop	0.0550	0.0852	0.1484	0.0812	0.0795	0.0824	0.0782	0.0799	0.0933	0.0446	0.0913	0.2186
	ItemPop	0.0511	0.0807	0.1428	0.0767	0.0749	0.0777	0.0697	0.0740	0.0895	0.0402	0.0856	0.2107
	CE-VAE	<u>0.2803</u>	<u>0.4104</u>	<u>0.5916</u>	<u>0.9418</u>	<u>0.9168</u>	<u>0.8820</u>	<u>0.9186</u>	<u>0.8760</u>	<u>0.8186</u>	0.1448	0.2683	0.4829
	CE-VNCF	0.2390	0.3221	0.4080	0.3414	0.3117	0.2690	0.3145	0.2633	0.2026	<b>0.1966</b>	<b>0.3263</b>	<b>0.4962</b>
	M&Ms-VAE (Ours)	<b>0.2817</b>	<b>0.4147</b>	<b>0.5960</b>	<b>0.9463</b>	<b>0.9237</b>	<b>0.8885</b>	<b>0.9243</b>	<b>0.8861</b>	<b>0.8256</b>	0.1457	0.2722	<u>0.4869</u>
CDs&Vinyl	UserPop	0.1028	0.1285	0.1869	0.0910	0.0807	0.0700	0.0860	0.0636	0.0595	0.1157	0.1704	0.3392
	ItemPop	0.1109	0.1357	0.1935	0.0928	0.0833	0.0716	0.0929	0.0657	0.0606	0.1288	0.1825	0.3499
	CE-VAE	<u>0.5243</u>	<u>0.6468</u>	<u>0.7454</u>	<u>0.6441</u>	<u>0.5609</u>	<u>0.4575</u>	<u>0.5564</u>	<u>0.4324</u>	<u>0.3040</u>	0.4795	<u>0.6808</u>	<u>0.8757</u>
	CE-VNCF	0.4590	0.5338	0.5860	0.3657	0.2981	0.2258	0.2893	0.2010	0.1260	<b>0.5081</b>	0.6698	0.8127
	M&Ms-VAE (Ours)	<b>0.5447</b>	<b>0.6659</b>	<b>0.7628</b>	<b>0.6648</b>	<b>0.5779</b>	<b>0.4700</b>	<b>0.5777</b>	<b>0.4441</b>	<b>0.3091</b>	<u>0.5015</u>	<b>0.6996</b>	<b>0.8894</b>
Yelp	UserPop	0.0007	0.0009	0.0066	0.0009	0.0010	0.0016	0.0013	0.0009	0.0061	0.0007	0.0011	0.0129
	ItemPop	0.0008	0.0011	0.0073	0.0009	0.0011	0.0018	0.0015	0.0011	0.0065	0.0009	0.0013	0.0149
	CE-VAE	<u>0.1935</u>	<u>0.2763</u>	<u>0.3803</u>	<u>0.6653</u>	<u>0.6356</u>	<u>0.5916</u>	<u>0.6363</u>	<u>0.5876</u>	<u>0.5181</u>	<u>0.1017</u>	<u>0.1819</u>	<u>0.3080</u>
	CE-VNCF	0.0883	0.1164	0.1505	0.1195	0.1052	0.0901	0.1023	0.0848	0.0690	0.0779	0.1270	0.2027
	M&Ms-VAE (Ours)	<b>0.1949</b>	<b>0.2787</b>	<b>0.3837</b>	<b>0.6738</b>	<b>0.6428</b>	<b>0.5976</b>	<b>0.6434</b>	<b>0.5935</b>	<b>0.5229</b>	<b>0.1019</b>	<b>0.1834</b>	<b>0.3107</b>
Hotel	UserPop	0.0436	0.0681	0.1059	0.1091	0.1059	0.1054	0.1018	0.1036	0.1050	0.0265	0.0557	0.1120
	ItemPop	0.0483	0.0756	0.1152	0.1159	0.1137	0.1124	0.1108	0.1131	0.1111	0.0303	0.0633	0.1237
	CE-VAE	<u>0.2425</u>	<u>0.3521</u>	<u>0.4984</u>	<u>0.9389</u>	<u>0.9113</u>	<u>0.8638</u>	<u>0.9209</u>	<u>0.8629</u>	<u>0.7831</u>	0.1153	<u>0.2105</u>	<u>0.3704</u>
	CE-VNCF	0.1873	0.2527	0.3280	0.3975	0.3671	0.3230	0.3732	0.3154	0.2558	<b>0.1252</b>	0.2060	0.3230
	M&Ms-VAE (Ours)	<b>0.2500</b>	<b>0.3595</b>	<b>0.5054</b>	<b>0.9752</b>	<b>0.9393</b>	<b>0.8829</b>	<b>0.9498</b>	<b>0.8776</b>	<b>0.7895</b>	<u>0.1182</u>	<b>0.2131</b>	<b>0.3726</b>

### Top-K Explanation Performance.

We also compare M&Ms-VAE with the user and item-popularity baselines (Wu et al., 2019a) that predict the explanation through counting and ranking the frequency of keyphrases for the users (symmetrically, the items) in the training set. Among the recommender baselines, only CE-VAE and CE-VNCF produce an explanation alongside the recommendation. We report the following metrics: NDCG, MAP, Precision, and Recall at different Top-K.

Table 5.4 contains the results. Both popularity baselines clearly underperform, showing that the task is not trivial (see Table 5.1 for the number of keyphrases per dataset). Remarkably, the proposed M&Ms-VAE model significantly outperforms the CE-VNCF baseline by a factor of 2.5 on average and by approximately by 3.5 on MAP and Precision. Moreover, M&Ms-VAE retrieves 89% percent of relevant keyphrases within the Top-20 explanations on the CDs&Vinyl dataset.

We observe that CE-VAE performs similarly to M&Ms-VAE but still slightly underperforms by approximately 2% on average. Finally, we note that CE-VNCF achieves the best results in terms of Recall for the Beer dataset and Recall@5 for the CDs&Vinyl and Hotel datasets. Nevertheless, as seen in the recommendation performance, CE-VNCF significantly underperforms, highlighting the trade-off between recommendation and explanation.

### 5.5.4 RQ 2: Can M&Ms-VAE with the self-supervised critiquing objective enable multi-step critiquing?

We now determine whether M&Ms-VAE can enable multi-step critiquing and effectively re-rank recommended items.

#### Baselines.

**UAC** (Luo et al., 2020a) denotes uniform average critiquing, in which the user embedding and all critique embeddings are averaged uniformly. **BAC** (Luo et al., 2020a) is balanced average critiquing. It first averages the critique embeddings and then averages them again with the initial user embedding. **CE-VAE** and **CE-VNCF** were introduced in Section 5.5.3. During training, both learn an inverse feedback loop between a critique and the latent space. At inference, they average the original user embedding with the critique embedding. **LLC-Score** (Luo et al., 2020a) and **LLC-Rank** (Li et al., 2020) first extend the PLRec recommender system (Sedhain et al., 2016) to co-embed keyphrases into the user embedding space with a linear regression. Afterwards, the models apply a weighted average between the initial user embedding and each critique embedding; the weights are optimized in a linear programming formulation; LLC-Score uses a max-margin scoring-based objective and LLC-Rank a ranking-based objective. To limit computational complexity, the authors limit the number of constraints to the top-100 rated items. For a fair comparison, we also consider in M&Ms-VAE the top-100 rated items meeting the criteria for  $I^{+c}$  and  $I^{-c}$  for each critique  $c$ , although the computational time remains identical using the full sets.

#### User Simulation.

Similarly to prior work (Luo et al., 2020a; Li et al., 2020), we conduct a user simulation to assess each model’s performance in a multi-step conversational recommendation scenario. Concretely, the simulation considers all users and follows Algorithm 2 with the following differences:

1. we track the conversational interaction session of simulated users by selecting all target items from their **test** set,
2. the maximum allowed critiquing iterations is set to 10, and,
3. the conversation stops if the target item appears within the top-N recommendations on that iteration.

As in Li et al. (2020), we simulate a variety of user-critiquing styles. For each, the candidate keyphrases to critique are inconsistent (i.e., disjoint) with the target item’s known keyphrase list. We experiment with the following three methodologies:

1. **Random**: we assume the user randomly chooses a keyphrase to critique.

2. **Pop**: we assume the user selects a keyphrase to critique according to the general keyphrase popularity.
3. **Diff**: we assume the user critiques a keyphrase that deviates the most from the known target item description. To do so, we compare the top recommended items' keyphrase frequency to the target item's keyphrases and select the keyphrase with the largest frequency differential.

### Multi-Step Critiquing Performance.

Following Luo et al. (2020a); Li et al. (2020), we assess the models over all users and all items on the test set using the average success rate and session length as metrics. The former is the percentage of target items that successfully reach a rank within the Top-N, and the latter is the average length of these sessions (with a limit of 10 iterations). In our experiment framework, for each user and target, we sample alongside 299 unseen items. We believe that this setting reflects realistic scenarios, where users navigate through hundreds of items instead thousands.

The results for each dataset and keyphrase critiquing selection method are depicted in Figure 5.4. Overall, all models' performance is generally better on the Beer and CDs&Vinyl datasets due to the higher density in terms of the number of interactions. Generally, all models tend to find the target item within the Top-20 in more than half the time and under six turns. This highlights that in practice, users are likely to find the desired item in a limited amount of time.

Impressively, M&Ms-VAE clearly outperforms all the baselines on both metrics on the Beer, CDs&Vinyl, and Yelp datasets. On the Hotel dataset, the success rate is significantly higher for the Random and Pop cases and similar for the Diff case, whereas the session length is higher for the Pop and Diff selection methods. This is unsurprising, because the blending module is trained only once on the random keyphrase critiquing selection (i.e., no assumption on the user's behavior).

Although the simple self-supervision objective in M&Ms-VAE mimics only one-step random critiquing, we remark that the training strategy generalizes for multi-step critiquing and other keyphrase critiquing selection as well. This shows that M&Ms-VAE efficiently embeds the critique, thanks to the multimodal modeling.

We observe that the simple UAC and BAC methods perform similarly or better than CE-VNCF and CE-VAE. However, they are outperformed by LLC-Score, LLC-Rank, and M&Ms-VAE. These results confirm our observation in Section 5.3.3 that the critiquing objective introduces noise during training and does not accurately reflect the critiquing mechanism.

Finally, we note that LLC-Score performs similarly to LLC-Rank in most cases. When compared to M&Ms-VAE, both significantly underperform on both metrics in 10 out of 12 cases. This highlights the effectiveness of our proposed critiquing algorithm compared to linear aggregation methods.

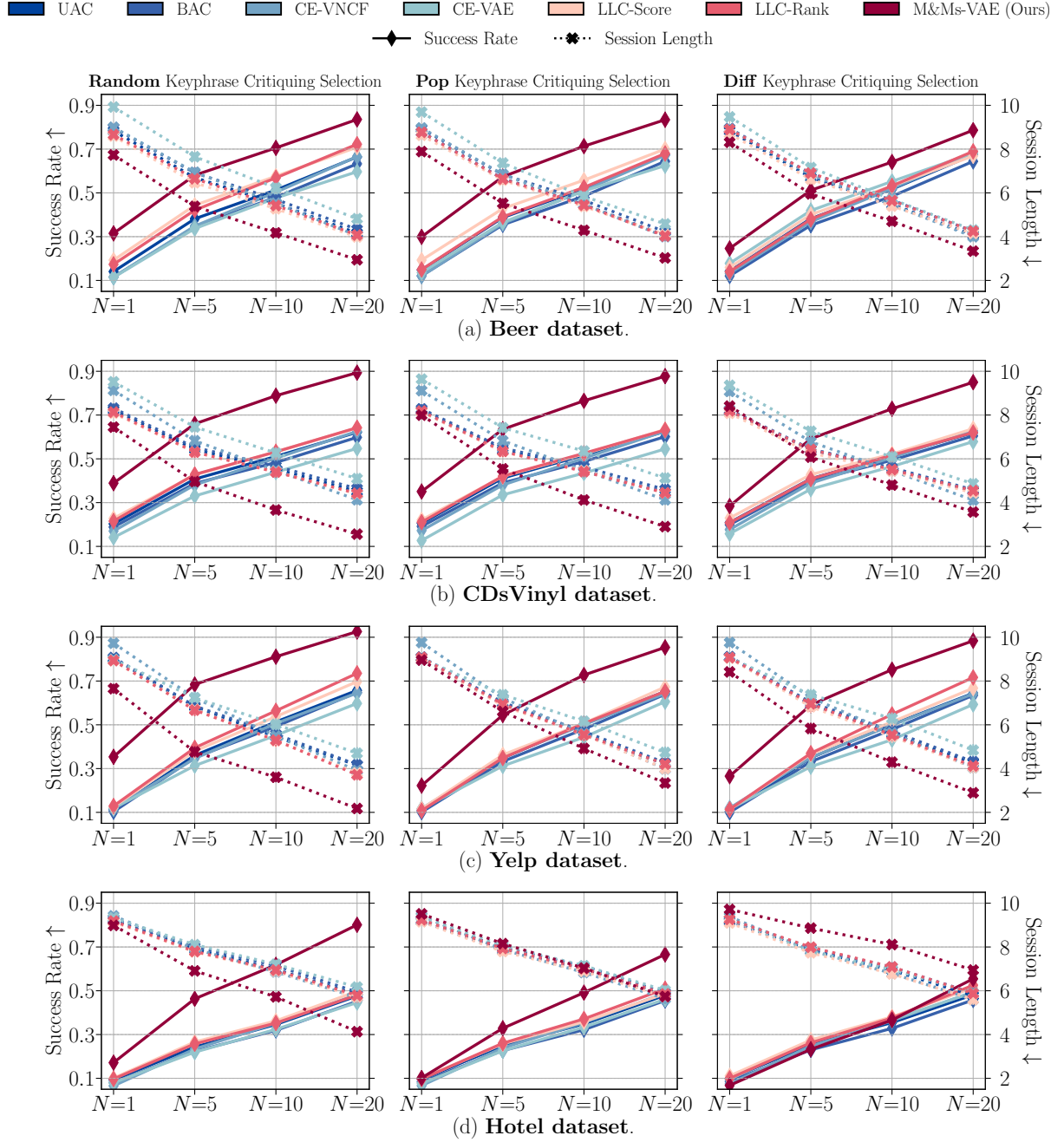


Figure 5.4: Multi-step critiquing performance after 10 turns of conversation. For each dataset and keyphrase critiquing selection method, we report the average success rate (left y-axis) and session length (right y-axis) at different Top-N with 95% confidence interval.

Finally, we replicate the experiment by using all the available items (see Table 5.1 for the sizes). When the evaluation is conducted on 300 items (see Figure 5.4), we see that users indeed find a specific item using our technique with a high success rate (i.e., an average success rate of 80%). However, in Figure 5.5 where thousands of items are available, the results show that

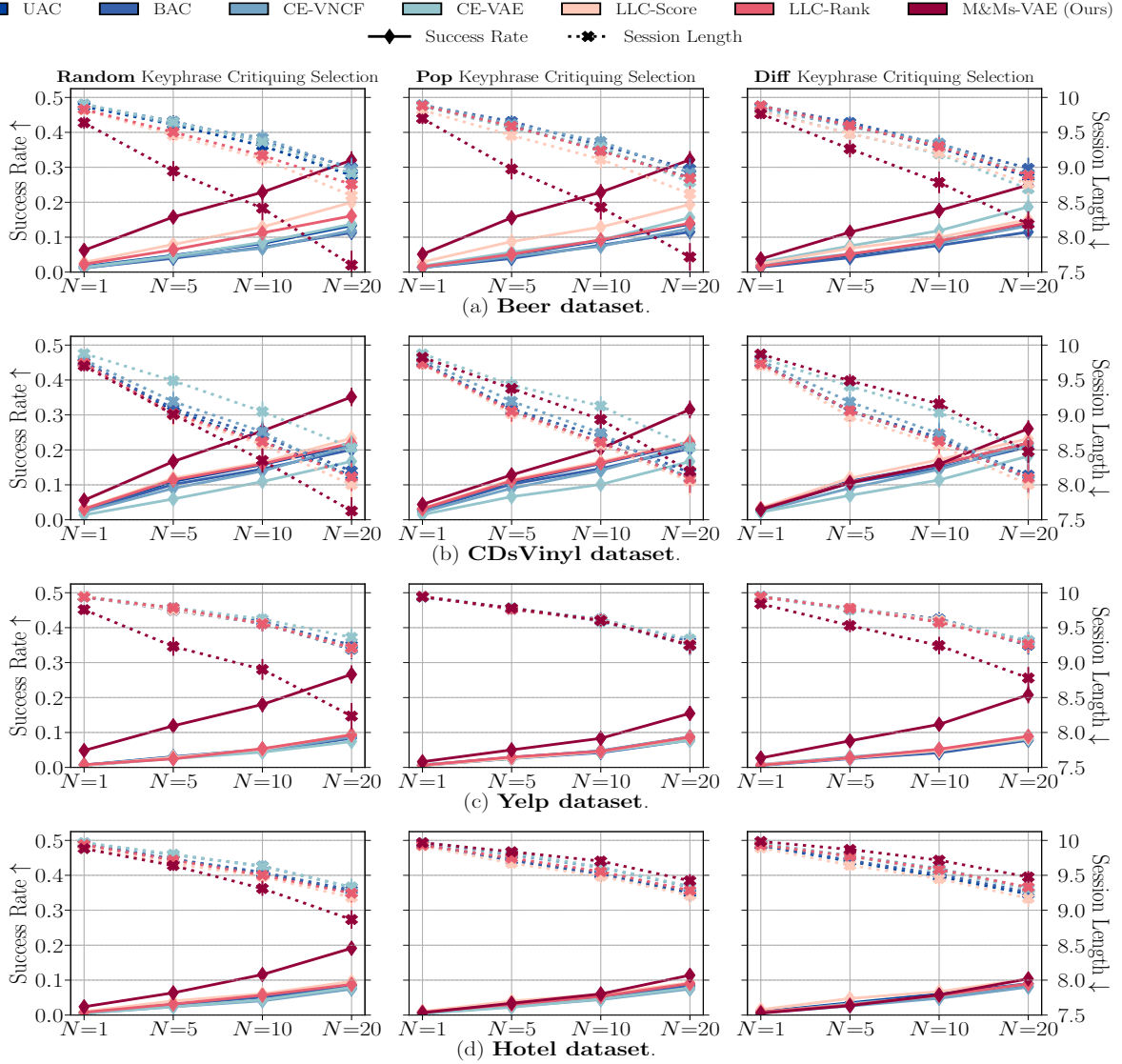


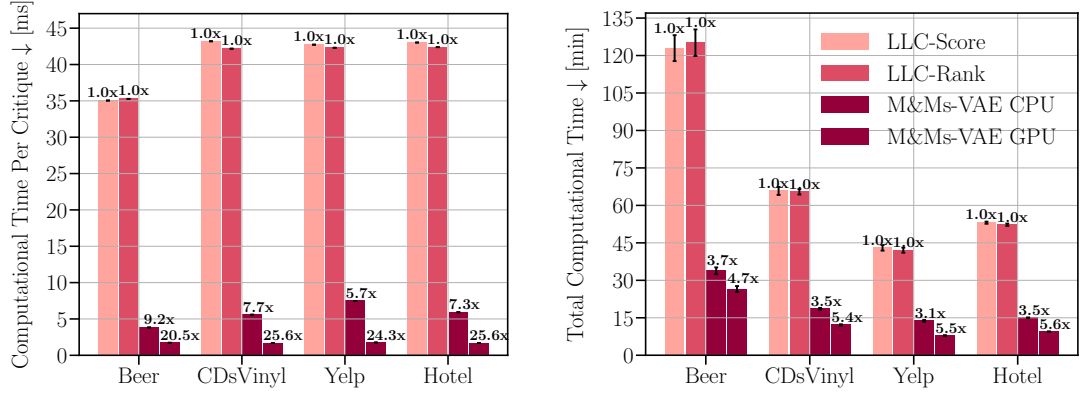
Figure 5.5: Multi-step critiquing performance after 10 turns of conversation **on all items**. For each dataset and keyphrase critiquing selection method, we report the average success rate (left y-axis) and session length (right y-axis) at different Top-N with 95% confidence interval.

current methods are not yet good enough to achieve similar results for such a large number. Nevertheless, M&Ms-VAE clearly outperforms other methods and still achieves an average success rate of 20%.

### 5.5.5 RQ 3: What is the critiquing computational time complexity for M&Ms-VAE compared to prior work?

Now, we aim to empirically determine how the critiquing in M&Ms-VAE compares to the best baselines in Section 5.5.4, LLC-Score and LLC-Rank, in terms of computational time. Because





(a) Average computational time of a single critique. (b) Average total computational time of the simulation.

Figure 5.6: Average time consumed for completing 10 runs for 1,000-user simulation after ten turns of conversation with 95% confidence intervals. LLC-Score and LLC-Rank cannot leverage GPUs; we thus report the performance of M&Ms-VAE on CPU and GPU. Additionally, we report the inference speedup compared to the slowest model.

the baselines can not leverage the GPU due to their optimization framework, we also run M&Ms-VAE on the CPU. We follow the same experiment settings as in Section 5.5.4 and limit ourselves to 1,000 users and the Random keyphrase critiquing selection method. All models process exactly 10 critiques for each user-item pair. We employ a machine with a 2.5GHz 24-core CPU, Titan X GPU, and 256GB memory (see Appendix F.4 for the detailed hardware).

Figure 5.6 shows the average computational time over 10 runs. Particularly, the Figure 5.6a denotes the critiquing computational time in milliseconds, and Figure 5.6b the overall simulation time in minutes. Impressively, we observe that the critiquing in M&Ms-VAE’s is approximately 7.5x faster on CPU and up to 25.6x on the GPU than LLC-Score and LLC-Rank. This shows that once the critiquing module of M&Ms-VAE is trained, which takes less than five minutes on the machine, the model achieves a lower latency (batch size of one). In Figure 5.6b, the overall simulation in M&Ms-VAE is at least 3.1x faster on CPU and approximately 5.3x faster on GPU. Finally, in real-life applications, we could leverage multiple users’ critiques simultaneously and increase the throughput by considering a larger batch size.

#### 5.5.6 RQ4: How does M&Ms-VAE perform under weak supervision; is the joint & cross inference coherent?

We first quantify the coherence of joint and cross generations of our M&Ms-VAE model. We denote three cases at test time:

1. only the user’s interactions are used, and the encoder is  $q_{\Phi_r}(\mathbf{z}_u|\mathbf{r}_u)$ ;
2. only the user’s keyphrase preference is used, and the encoder is  $q_{\Phi_k}(\mathbf{z}_u|\mathbf{k}_u)$ ; and
3. both are used with the encoder  $q_{\Phi}(\mathbf{z}_u|\mathbf{r}_u, \mathbf{k}_u)$ .

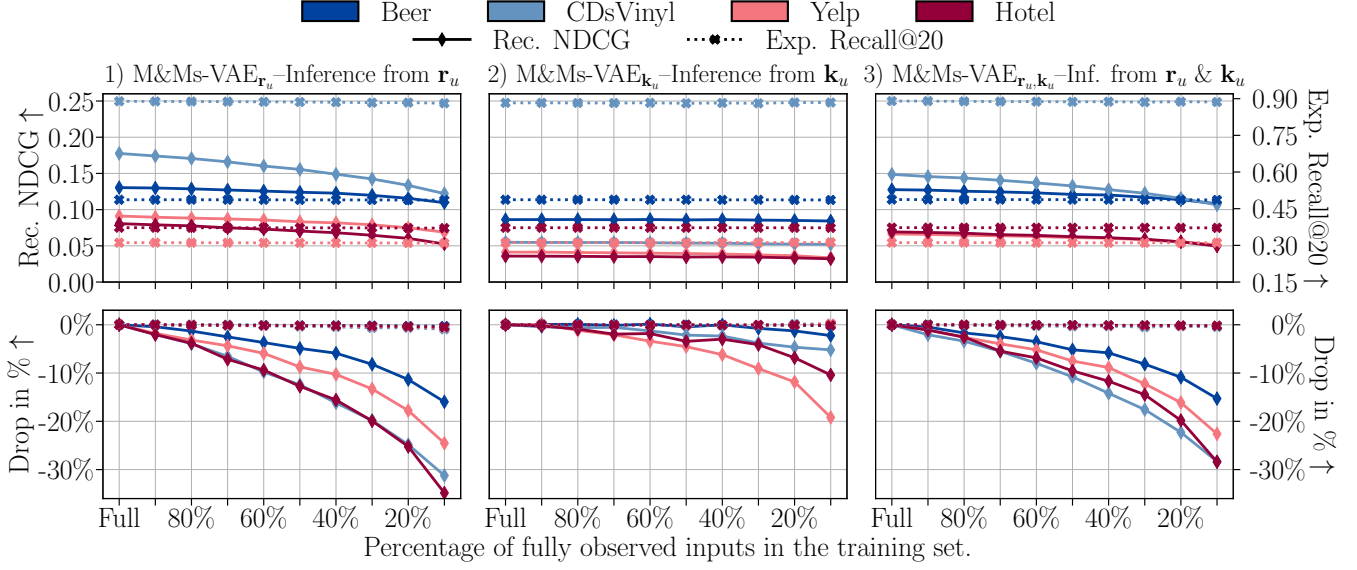


Figure 5.7: Recommendation (left y-axis) and explanation (right y-axis) results averaged on five runs with different combinations of modalities observed at inference. The x-axis denotes the percentage of fully observed inputs during training; the rest is partially observed.

Second, we simulate incomplete supervision by randomly selecting a subset of the training with fully observed samples. The other one is split into two even parts: the first includes only observed **r<sub>u</sub>** and the second **k<sub>u</sub>**. We retain the models and settings of Section 5.5.3.

Figure 5.7 shows the results averaged on five runs for the three cases at different levels of supervision. The top row presents the explanation and recommendation performance in terms of NDCG and Recall@20 (the model behavior is consistent across the other metrics). On the full datasets, we note that the explanation performance is comparable across the three variants and higher than those of the baselines in Table 5.4.

Regarding the recommendation performance, M&Ms-VAE<sub>r<sub>u</sub></sub> obtains the best results, followed closely by M&Ms-VAE<sub>r<sub>u</sub>,k<sub>u</sub></sub>, and M&Ms-VAE<sub>k<sub>u</sub></sub> clearly underperforms. This aligns with the observations in Sachdeva and McAuley (2020); Chen et al. (2018a): recommender systems are limited if they use only review text as input, and not all reviews can be useful. Nevertheless, compared to Table 5.3, M&Ms-VAE<sub>k<sub>u</sub></sub> always achieves better recommendation performance than the popularity baseline, and it performs better than AutoRec and CDAE on two datasets. The bottom row of Figure 5.7 show the relative drop in performance on both metrics. The explanation performance seems unaffected by the sparsity, showing that the explanation task remains simple in comparison with the recommendation task. Remarkably, with only 50% fully observed inputs and the rest partially observed, the recommendation performance of M&Ms-VAE<sub>r<sub>u</sub></sub> and M&Ms-VAE<sub>r<sub>u</sub>,k<sub>u</sub></sub> is decreased by only 9% on average. More so, with 90% partial observations, the model can still achieve more than 70% of its performance quality on the full datasets. Finally, these results emphasize that M&Ms-VAE can effectively learn the joint distribution even in a weakly supervised setting.

## 5.6 Conclusion

Recommendations can have much more impact if they are supported by explanations that can be critiqued. Previous research has developed methods that either perform poorly in multi-step critiquing or suffer from computational inefficiency at inference. In this chapter, we presented M&Ms-VAE, a novel variational autoencoder for recommendation and explanation that treats the user preference and keyphrase usage as different observed variables. Additionally, we proposed a strategy that mimics weakly supervised learning and trains the inference networks jointly and independently.

Our second contribution is a new critiquing module that leverages the generalizability of M&Ms-VAE to embed the user preference and the critique. With a self-supervised objective and a synthetic dataset, we enable multi-step critiquing in M&Ms-VAE. Experiments on four datasets show that M&Ms-VAE

1. is the first model to obtain substantially better recommendation, explanation, and multi-critiquing performance,
2. processes critiques up to 25.6x faster than previous state-of-the-art methods, and
3. produces coherent joint and cross generation, even under weak supervision.



## 6 Conclusion

This dissertation’s initial motivation comes from the observation that machine learning algorithms pervade our lives. Because more data are available than ever before, automated predictions accompanied by explanations can carry much more impact, can increase transparency, and may even be necessary. Nevertheless, it remains challenging to interact with such predictions when they are unsatisfying or wrong. Consequently, end users might feel frustrated with or, worse, experience distrust for or reject the model predictions. Although many interpretable techniques (e.g., Ribeiro et al., 2016; Serrano and Smith, 2019; Li et al., 2016b) and inherently interpretable models (e.g., Alvarez-Melis and Jaakkola, 2018; Koh et al., 2020; Lei et al., 2016) exist, the explanations generated are mostly informative and provided “as it is”. However, it has been shown that users prefer the opportunity to provide feedback (i.e., tend to critique the model’s outputs or its explanations) that will improve the model (Amershi et al., 2014; Wall et al., 2019; Smith-Renner et al., 2020). This preference highlights the growing need for users to better understand the underlying computational processes and control their actions better.

Part I of this dissertation examines the problem of **explanation generation**: how can we infer high-quality explanations from text documents? We base our solution on the paradigm of selective rationalization, that is, finding relevant subsets of the input text (called a rationale) that suffice on their own to yield the same prediction. Our key motivation arises from the limitations of rationales. Current rationalization models generate only one rationale: they strive for one hard (binary) overall selection to explain one outcome. This limitation imposes multiple constraints, such as assuming some priors in the data distribution, which do not reflect realistic settings, and makes it difficult to capture ambiguities in the text. We claim that useful rationales can be multi-faceted, thus involving support for different outcomes. For example, clinicians justify medical diagnoses and treatments of patients, based on multiple factors: symptoms, prior events, and preconditions. In the case of user reviews, users justify their overall feedback with multiple explanations - one for each aspect they deem important.

In Chapter 2, we propose the notion of multi-dimensional rationales and a multi-task learning approach in which rationales are learned in an unsupervised manner. To the best of our

knowledge, our study is the first to neither assume any prior on the data distribution nor train, tune, or maintain multiple models, one for each label to explain (e.g., one label per subrating). Therefore, our model can be deployed in realistic settings and learn soft multi-dimensional rationales in one training. We also empirically demonstrate that interpretability does not come at the cost of reduced performance and that the resulting explanations are more accurate and coherent than those generated by other methods.

In Chapter 3, we extend the multi-dimensional idea and generalize our model by assuming that only one label is observed (e.g., only the overall rating). Inspired by the role of concept-based thinking in human reasoning (Armstrong et al., 1983; Tenenbaum, 1999), we identify latent concepts and explain the outcome with a linear aggregation of concepts. Our study makes pioneering inroads into the extraction of multi-dimensional explanations and latent concepts for selective rationalization. It extends the application to cases in which multiple labels are not available at a large scale (e.g., on online platforms, subratings are optional whereas the overall rating is always mandatory). We empirically show that our model generates concepts that align with human rationalization while using only the overall label. Further, it outperforms state-of-the-art methods trained on each aspect label independently. Finally, both of the approaches we propose are trivially extendable for images as input data.

Part II of this dissertation focuses on extending modern neural recommendation systems with explanations and interactive **critiquing**, which results in conversational explainable recommendation systems. Interactions are naturally present in recommendation systems, that constitute an impactful real-life application with hundreds of millions of predictions made daily.

Studies of explainable recommendation systems have shown that users prefer justifications alongside recommended items that are personalized and written in natural language (i.e., that could have been written by other humans) (Kunkel et al., 2019; Chang et al., 2016; Wilkinson et al., 2021). However, there are no ground-truth datasets specifying what “good” justifications are. Prior studies have employed users’ reviews to capture their preferences and writing styles (e.g., Ni and McAuley, 2018; Li and Tuzhilin, 2019; Dong et al., 2017). From past reviews, they generate synthetic ones that serve as personalized explanations of ratings given by users. However, many reviews are noisy because they partly describe experiences or endorsements. Ni et al. (2019) proposed a nonscalable pipeline for identifying justifications from reviews and asked humans to annotate them. Chen et al. (2019b, 2020) positioned the justification as the first sentence. However, the concepts of justification offered by these studies were ambiguous, and they assumed that a review contains only one justification. It is thus nontrivial to identify meaningful justifications within reviews.

Based on the models developed in Part I, in Chapter 4, we address the explanation generation challenge by introducing a scalable and data-driven justification extraction system, with no prior limits imposed on their number or structure. This approach relies on the faithfulness property of rationales, which is naturally compatible with recommendation: the data include

user ratings and reviews. Our study is the first to introduce personalized explanations based on rationales and to show that they are significantly preferred by human users for recommendation in multiple domains over those produced by extant state-of-the-art techniques.

After nearly a decade in which critiquing approaches received little attention, Wu et al. (2019a); Luo et al. (2020b,a); Li et al. (2020) introduced a collaborative filtering recommender with keyword-based explanations and an embedding-based critiquing method. These methods allow users to critique the recommendation using arbitrary languages; a set of attributes is mined from reviews, and the users can interact with them. However, isolated keywords are not the best type of explanation for users, and it is not trivial to co-embed item critiques with the latent user preferences. Furthermore, Luo et al. (2020b,a); Li et al. (2020) showed that previous models suffer from computational inefficiency and an inherent trade-off between recommendation, explanation, and critiquing performance.

In the second part of Chapter 4, we propose a transformer-based model (Vaswani et al., 2017) that generates abstractive natural language justifications as well as keyword-based explanations that accompany the recommendation. A crucial question remains: how can we critique the suggested items with their explanations made by the recommender system? We incorporate critiquing via a novel unsupervised gradient-based technique that allows users to interact with the justifications. Keyword-based explanations are the key to connecting the justifications and the model’s inner workings. We demonstrate empirically that our system successfully adapts to the preferences expressed in multi-step critiquing and generates consistent explanations without decreasing recommendation performance. Finally, to investigate the effects of different recommendation interfaces on users, we propose an online model-agnostic platform to foster research on recommendation with explanations and critiquing. Our interfaces allow to test such systems on a real use case and highlight some limitations of these approaches.

Finally, Chapter 5 presents a novel variational autoencoder (Kingma and Welling, 2014) for recommendation based on multimodal modeling assumptions and self-supervised training. It treats the user preference and keyphrase usage as different observed variables. Our critiquing module, which leverages the generalizability of our model to embed the user preference and the critique, is an important innovation. With a self-supervised ranking objective and an automatically generated synthetic dataset, we enable fast and efficient multi-step critiquing. Using four real-world, publicly available datasets, we show that compared to extant state-of-the-art models, our model not only exceeds or matches their performance in terms of recommendation, explanation, and multi-step critiquing, but also processes the critiques up to 25.6x faster. This may accelerate the adoption and deployment of conversational explainable recommendation systems into a variety of online platforms.

## 6.1 Future Directions

Our study points to several potential directions for future research. We outline the most promising ones below.

1. **Decreasing neural hallucinations.** Due to the recent progress in natural language generation, models can now generate syntactically and grammatically correct sentences (Radford et al., 2019; Brown et al., 2020). As is often the case in neural text generation, however, the justifications generated may suffer from factual inconsistency (Rohrbach et al., 2018; Holtzman et al., 2020). For example, our system might recommend a hotel with a justification mentioning “Lugano” even though the hotel is located in Madrid. Although such justifications might be factually correct in some cases, they usually are not. A potential solution for this problem is to enrich the neural decoder with commonsense knowledge to verify the consistency of generated facts.
2. **Treating critiques as harder constraints.** In Chapters 4 and 5, we co-embed the language-based feedback (i.e., critique) and user preferences in the same space. Then, we combine each representation through a blending function to infer the final latent representation. This blending function plays a key role in satisfying the constraints based on the current and previous critiques. However, we have little control over it: there is no guarantee that the function will produce a latent representation that fully integrates and satisfies the critique. A promising direction is neural-symbolic reasoning over knowledge graphs. User behaviors and preferences, item attributes, and critiques can be represented as sets of symbolic rules derived from the knowledge graph. Nevertheless, the knowledge graph is domain dependent, which limits its generalization unless it is built automatically.
3. **Choosing better keywords to critique and allowing categorical critiquing.** Another limitation of our system is the frequency-based approach with some postprocessing to mining keywords. Although our method is automatic and allows for a richer interaction space than does a set of fixed item attributes, meaningful keywords for the recommendation do not necessarily imply that the keywords are helpful for end users. We believe improvements can be made along two axes: the keywords’ interactions and semantic representation. Regarding keywords’ interactions, it is unclear, for example, what it means to critique the keyword “breakfast” positively or negatively. Currently, we assume only a binary constraint: the presence or absence of breakfast service in an establishment. It would be more useful to treat the constraint as a categorical variable (Balog et al., 2021; Louis et al., 2020). Each possible value would also be mined from the context in which keywords appear (e.g., buffet or continental for the keyword “breakfast”). This could be achieved through aspect- and opinion-term-extraction techniques, and the critique encoding would be extended from a binary representation to an ordinal one. Regarding semantic representation, we could extract keywords based on their surface forms and their semantic representation (e.g., Bennani-Smires et al., 2018). We could identify clusters with their representative keywords while controlling diversity. Thus, the keywords’ latent representations would embed an actual meaning. As a result, the model would better understand the meaning of a critique, because the keywords and critiques would interact with the same latent space.
4. **Relaxing the constraints of latent concepts.** In Chapter 3, we generalize the approached



described in Chapter 2 by observing only one variable. However, this generalization comes at the cost of two constraints specified beforehand: the number of concepts and their length. Early experiments without these constraints produced significantly worse results. We suspect that the predictor overfits the rationales found by the generator. Thus, both might be trapped in a suboptimal equilibrium. This phenomenon – called model interlocking – was recently confirmed in Yu et al. (2021). One means of alleviating this problem would be to pretrain the generator to extract more diverse and semantically meaningful text snippets via self-contrastive learning-based methods or topic-modeling-based approaches. In this way, the joint optimization might start with a better initialization and more successfully identify the text snippets that constitute potential rationales (i.e., meaningful for the prediction).

The research directions proposed above can be combined to make even more significant improvements in the quality of justifications, recommendations, and user experience.



## Appendices **Part**



# A Complementary Results

## A.1 Inferred Concepts by ConRAT

The following results complements Section 3.5.5.

### A.1.1 Objective Evaluation

The results for the concept length  $\ell = 5$  is shown in Table A.1.

Table A.1: Objective performance of rationales for the multi-aspect beer reviews with the concept length set to five. ConRAT only uses the overall rating and does not have access to the other aspect labels. All baselines are trained separately on each aspect label. **Bold** and underline denote the best and second-best results, respectively.

Model	Acc.	Average			Appearance			Aroma			Palate			Taste			Overall		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
RNP	80.8	41.3	10.4	16.4	<u>50.9</u>	<b>13.3</b>	<b>21.1</b>	<b>43.2</b>	<b>12.7</b>	<b>19.7</b>	<u>27.1</u>	<u>10.0</u>	<u>14.5</u>	5.5	0.59	1.07	80.0	15.3	25.7
RNP-3P	81.5	32.9	6.9	11.2	35.1	7.3	12.1	25.6	7.2	11.3	17.0	5.2	8.0	28.6	4.0	7.1	58.2	10.5	17.8
Intro-3P	<u>84.6</u>	29.8	7.0	11.3	47.3	12.4	19.7	35.4	9.9	15.5	9.7	2.8	4.3	24.3	3.8	6.6	32.4	6.3	10.6
InvRAT	83.6	<u>46.4</u>	<b>11.4</b>	<b>18.1</b>	<b>51.0</b>	<u>13.1</u>	<u>20.8</u>	<u>40.6</u>	<u>11.9</u>	<u>18.4</u>	<b>32.0</b>	<b>11.8</b>	<b>17.2</b>	<u>36.1</u>	<u>5.6</u>	<u>9.6</u>	72.5	14.7	24.4
ConRAT <sup>†</sup>	<b>90.4</b>	<b>46.6</b>	<u>10.9</u>	<u>17.5</u>	47.2	12.4	19.6	26.9	7.1	11.3	26.6	9.2	13.7	<b>39.2</b>	<b>6.2</b>	<b>10.8</b>	<b>93.1</b>	<b>19.5</b>	<b>32.21</b>

\* The model is only trained on the overall label and does not have access to the other ground-truth labels.

### A.1.2 Subjective Evaluation

All results (for the joint, the aspect, and the polarity accuracy) are shown in Figure A.1. In total, we used 7,500 samples ( $100 \times 5 \times 5 \times 3$ ).

We also studied the error rates on each aspect. The Aroma and Palate aspects cause the highest error for all models. One possible reason is that users confuse these with the aspect Taste, hence their high correlations in rating scores (Antognini et al., 2021b).

## Appendix A. Complementary Results

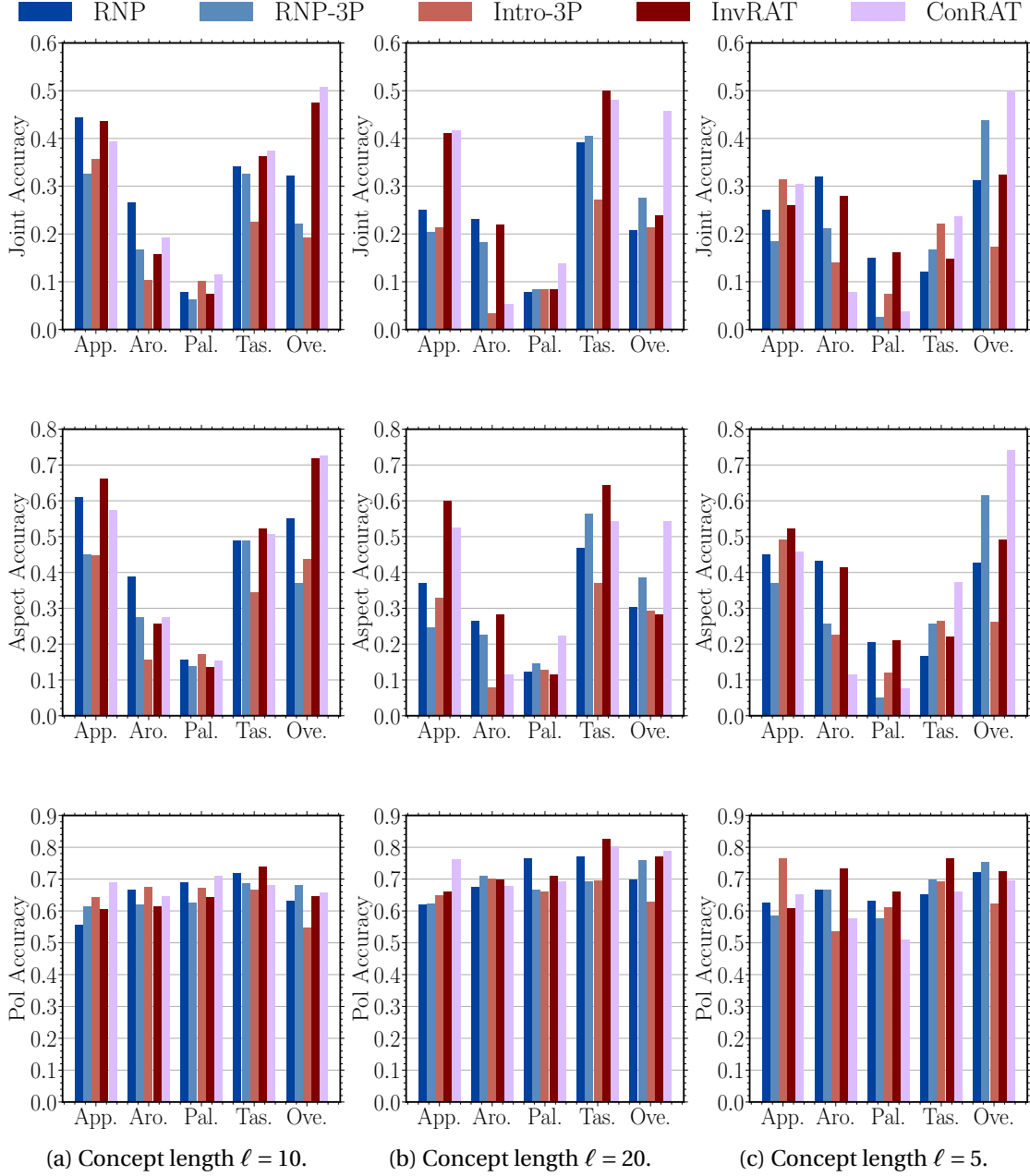


Figure A.1: Subjective performance per aspect of rationales for the multi-aspect beer reviews.

## A.2 Full Natural Language Explanations Results

We also compare T-RECS with more models than these of Section 4.6.3: LexRank (Erkan and Radev, 2004), NRT (Li et al., 2017), Item-Rand, Ref2Seq Top-k, and ACMLM (Ni et al., 2019). LexRank is a unsupervised multi-document summarizer that selects an unpersonalized justification among all historical justifications of an item. NRT generates an explanation based on rating and the word distribution of the review. Item-Rand is an unpersonalized baseline

## A.2 Full Natural Language Explanations Results

Table A.2: Performance of the generated personalized justifications on automatic evaluation.

	Model	B-1	B-2	B-3	B-4	R-1	R-2	R-L	BERT <sub>Score</sub>	PPL↓	R <sub>KW</sub>	R <sub>Sent</sub>
Hotel	Item-Rand	11.50	2.88	0.91	0.32	12.65	0.87	9.75	84.20	-	6.92	56.88
	LexRank	12.12	3.31	1.10	0.41	14.74	1.16	10.61	83.91	-	10.32	58.51
	ExpansionNet	4.03	1.95	1.01	0.53	34.22	9.65	6.91	74.81	28.87	60.09	61.38
	NRT	17.00	6.51	3.06	1.51	18.21	2.88	16.08	86.31	29.75	11.44	64.46
	DualPC	18.91	6.88	3.18	1.53	20.12	3.08	16.73	86.76	28.99	13.12	63.54
	CAML	10.93	4.11	2.09	1.13	15.44	2.37	16.67	87.77	29.10	16.17	65.14
	Ref2Seq	17.57	7.03	3.44	1.77	19.07	3.43	16.45	86.74	29.07	13.19	64.40
	Ref2Seq Top-k	12.68	3.46	1.11	0.40	12.67	0.95	10.30	84.29		6.38	58.11
	AP-Ref2Seq	32.04	19.03	11.76	7.28	38.90	14.53	33.71	88.31	21.31	90.20	69.37
	ACMLM	8.60	2.42	1.12	0.62	9.79	0.55	7.23	81.90	-	13.24	60.00
	T-RECS (Ours)	<b>33.53</b>	<b>19.76</b>	<b>12.14</b>	<b>7.47</b>	<b>40.29</b>	<b>14.74</b>	<b>34.10</b>	<b>90.23</b>	<b>17.80</b>	<b>93.57</b>	<b>70.12</b>
Beer	Item-Rand	10.96	3.02	0.91	0.29	10.28	0.75	8.25	83.39	-	6.70	99.61
	LexRank	12.23	3.58	1.12	0.38	13.81	1.16	9.90	83.42	-	10.79	99.88
	ExpansionNet	6.48	3.59	2.06	1.22	<b>54.53</b>	18.24	9.68	72.32	22.28	82.49	<b>99.99</b>
	NRT	18.54	8.53	4.40	2.43	17.46	3.61	15.56	85.26	21.22	11.43	<b>99.99</b>
	DualPC	18.38	8.10	3.95	2.08	17.61	3.38	14.68	85.49	21.15	10.60	<b>99.99</b>
	CAML	12.94	6.5	3.80	2.43	14.63	2.43	14.99	85.96	21.29	10.18	<b>99.99</b>
	Ref2Seq	18.75	9.47	5.51	3.51	18.25	4.52	15.96	85.27	22.34	12.10	<b>99.99</b>
	Ref2Seq Top-k	13.92	5.02	2.10	1.01	12.36	1.50	10.52	84.14		8.51	99.83
	AP-Ref2Seq	44.84	30.57	21.68	15.89	51.38	23.27	46.50	91.35	12.07	91.52	<b>99.99</b>
	ACMLM	7.76	2.54	0.91	0.34	8.33	0.87	6.17	80.94	-	10.33	<b>99.99</b>
	T-RECS (Ours)	<b>46.50</b>	<b>31.56</b>	<b>22.42</b>	<b>16.54</b>	53.12	<b>23.86</b>	<b>47.20</b>	<b>91.50</b>	<b>10.24</b>	<b>94.96</b>	<b>99.99</b>

which outputs a justification randomly from the justification history  $J^i$  of item  $i$ . Ref2Seq Top-k is an extension of Ref2Seq, where we explore another decoding strategy called Top-k sampling (Radford et al., 2019), which should be more diverse and suitable on high-entropy tasks (Holtzman et al., 2020). Finally, ACMLM is an aspect conditional masked language model that randomly chooses a justification from  $J^i$  (similar to Item-Rand) and then iteratively edits it into new content by replacing random words. We also include more metrics and R<sub>Sent</sub>, which computes the percentages of generated justifications sharing the same polarity as the targets according to a sentiment classifier<sup>1</sup>.

The complete results are presented in Table A.2. Interestingly, Item-Rand performs closely to LexRank: the best justification, according to LexRank, is slightly better than a random one. On the other hand, ACMLM edits the latter by randomly replacing tokens with the language model but produces poor quality justification, similarly to Ni et al. (2019). Finally, we also observe that the polarities of the generated justifications for beers match nearly perfectly, unlike in hotels where the positive and negative nuances are much harder to capture.

<sup>1</sup>We employ the sentiment classifiers trained jointly with Multi-Target Masker of Chapter 2, used to infer the *markers* from which the justifications are extracted.

### A.3 Full Keyphrase Explanation Results

Table A.3 contains more results to the keyphrase explanation quality experiment (Section 4.6.3).

Table A.3: Performance of personalized keyphrase explanation quality.

	Model	NDCG@N			MAP@N			Precision@N			Recall@N		
		N=5	N=10	N=20	N=5	N=10	N=20	N=5	N=10	N=20	N=5	N=10	N=20
Hotel	UserPop	0.2625	0.3128	0.3581	0.2383	0.1950	0.1501	0.1890	0.1332	0.0892	0.2658	0.3694	0.4886
	ItemPop	0.2801	0.3333	0.3822	0.2533	0.2083	0.1608	0.2041	0.1431	0.0959	0.2866	0.3961	0.5245
	E-NCF	0.2901	0.3410	0.3889	0.2746	0.2146	0.1618	0.1943	0.1366	0.0919	0.2746	0.3802	0.5057
	CE-NCF	0.1929	0.2286	0.2634	0.1825	0.1432	0.1085	0.1290	0.0918	0.0631	0.1809	0.2548	0.3469
	E-VNCF	0.2902	0.3441	0.3925	0.2746	0.2158	0.1634	0.1947	0.1391	0.0932	0.2746	0.3860	0.5132
	CE-VNCF	0.1727	0.2289	0.2761	0.1530	0.1336	0.1115	0.1275	0.1071	0.0767	0.1795	0.2965	0.4200
	T-RECS (Ours)	<b>0.3158</b>	<b>0.3763</b>	<b>0.4319</b>	<b>0.2919</b>	<b>0.2356</b>	<b>0.1807</b>	<b>0.2223</b>	<b>0.1581</b>	<b>0.1068</b>	<b>0.3109</b>	<b>0.4358</b>	<b>0.5812</b>
Beer	UserPop	0.2049	0.2679	0.3357	0.2749	0.2404	0.2014	0.2366	0.1901	0.1445	0.1716	0.2767	0.4207
	ItemPop	0.1948	0.2495	0.3131	0.2653	0.2291	0.1894	0.2267	0.1759	0.1342	0.1618	0.2529	0.3886
	E-NCF	0.1860	0.2485	0.3158	0.2488	0.2204	0.1877	0.2162	0.1789	0.1389	0.1571	0.2618	0.4040
	CE-NCF	0.1471	0.1922	0.2422	0.1967	0.1721	0.1446	0.1687	0.1363	0.1050	0.1227	0.1974	0.3033
	E-VNCF	0.1763	0.2362	0.3055	0.2389	0.2097	0.1797	0.2031	0.1696	0.1356	0.1471	0.2478	0.3943
	CE-VNCF	0.1512	0.2025	0.2595	0.1987	0.1784	0.1532	0.1774	0.1475	0.1155	0.1293	0.2146	0.3352
	T-RECS (Ours)	<b>0.2394</b>	<b>0.3163</b>	<b>0.3946</b>	<b>0.3127</b>	<b>0.2799</b>	<b>0.2369</b>	<b>0.2800</b>	<b>0.2284</b>	<b>0.1717</b>	<b>0.2048</b>	<b>0.3320</b>	<b>0.4970</b>

### A.4 Multi-Step Critiquing - Additional Metrics for T-RECS

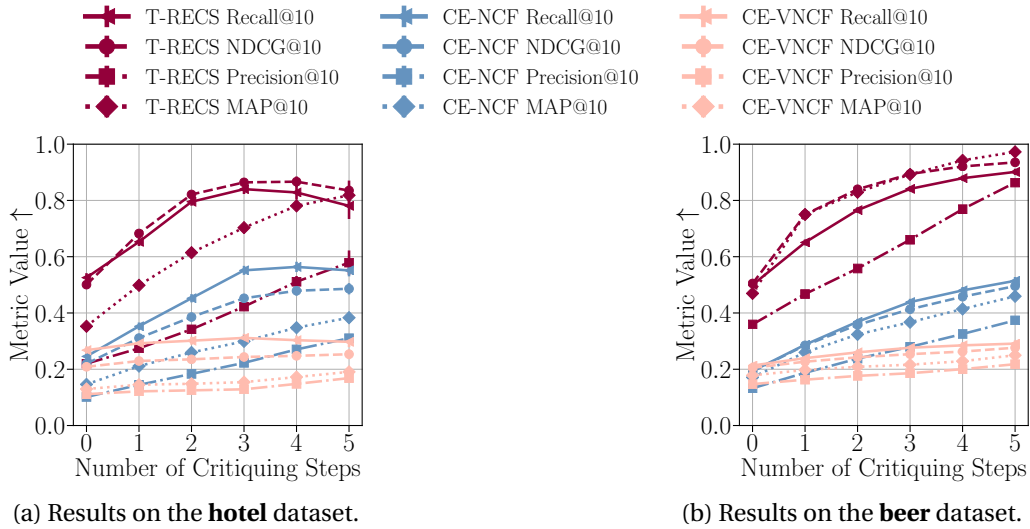


Figure A.2: Multi-step critiquing performance. Keyphrase prediction over multi-step critiquing in terms of Recall@10, NDCG@10, Precision@10, and MAP@10 with 95% confidence interval.

a) Results on the hotel dataset, b) on the beer dataset.



## B Processing

### B.1 Processing & Filtering *Markers*

The method described in Antognini et al. (2021b) extracts, most of the time, *markers* that consist of long, continuous spans of words. However, sometimes, the *markers* are too short because some reviews do not include enough words to justify a certain aspect rating, or the *markers* stop in the middle of a sentence. Although both are theoretically not wrong, we aim to create fluent and grammatically correct justifications. To this end, we exploit the constituency parse tree to ensure that *markers* are noun/verb phrases. We apply the following steps to the entire set of reviews for each dataset:

1. Compute the constituency parse tree of each review;
2. For each noun and adjective node in the constituency parse tree of a *marker*, if the parent node is a verb or noun phrase, we add its children to the *marker*. We follow the rules in Giannakopoulos et al. (2017);
3. Filter out *markers* having less than four tokens or including first and third-person pronouns.

In cases where faceted ratings are not available at large cases, Mukherjee and Awadallah (2020); Niu et al. (2020) proposed elegant solutions to infer them from 20 or fewer samples. If faceted ratings are not present, one can use the concept rationalizer of Antognini and Faltings (2021b) to infer *markers*.

### B.2 Extracting Keyphrase for T-RECS

None of our datasets contains initially preselected keyphrases. We extract 200 keyphrases from the *markers* used to model the user and item profiles. They serve as a basis for the explanation and the critiquing. Table B.1 shows some keyphrases for each dataset. We apply the following processing for each dataset, similarly to Wu et al. (2019a):

## Appendix B. Processing

Table B.1: Some keyphrases mined from the inferred *markers*. We grouped them by aspect for a better understanding.

Dataset	Aspect	Keyphrases
Hotel	Service	bar, lobby, housekeeping, guest
	Cleanliness	carpet, toilet, bedding, cigarette
	Value	price, wifi, quality, motel, gym
	Location	airport, downtown, restaurant shop
	Room	bed, tv, balcony, fridge, microwave
Beer	Appearance	golden, dark, white, foamy
	Aroma	fruit, wheat, citrus, coffee
	Palate	creamy, chewy, syrupy, heavy
	Taste	bitter, sweet, balanced, nutty

1. Group by aspect *markers* from all reviews. The aspect sets come from faceted ratings<sup>1</sup>;
2. For each group of *markers*:
  - Tokenize and lemmatize the entire set of *markers*;
  - Extract unigram lists of high-frequency noun and adjective phrases;
  - Keep the top- $k$  most likely unigrams;
3. Represent each review as a binary vector indicating the presence of keyphrases.

Another possibility to extract keywords from reviews is to leverage Microsoft Concept Graph as in Chen et al. (2019b, 2020). However, the API is limited to single instance conceptualization.

### B.3 Addressing Users without Reviews in T-RECS

The cold-start problem is not particular to our method but a general problem in recommendation. However, our system could infer  $\gamma^u$  and  $\beta^u$  for users without reviews and then computes the initial latent representation  $\mathbf{z}$  as in Section 4.5. We propose the following strategies:

1. **Users with ratings but no reviews:** we could leverage collaborative filtering techniques to identify users with similar ratings and build an initial representation.
2. **New users:** following the observation of Zhang et al. (2014); Musat and Faltings (2015): “users write opinions about the topics they care about”, we could ask new users to write about the different aspects they deem important. Their initial representation is an aggregation of other similar users. Another option is to ask users to select items they like based on a subset of items and build a profile from these preferences (see above).

---

<sup>1</sup>For the hotel reviews: service, cleanliness, value, location, and room. For beer reviews: appearance, smell, mouthfeel, and taste.

# C Visualizations

## C.1 Extracted Topic Words per Aspect from Rationales

For each model, we computed the probability distribution of words per aspect by using the induced sub-masks  $M_{a_1}, \dots, M_{a_A}$  or attention values. Given an aspect  $a_i$  and a set of top- $N$  words  $w_{a_i}^N$ , the Normalized Pointwise Mutual Information (Bouma, 2009) coherence score is:

$$\text{NPMI}(w_{a_i}^N) = \sum_{j=2}^N \sum_{k=1}^{j-1} \frac{\log \frac{P(w_{a_i}^k, w_{a_i}^j)}{P(w_{a_i}^k)P(w_{a_i}^j)}}{-\log P(w_{a_i}^k, w_{a_i}^j)} \quad (\text{C.1})$$

Top words of coherent topics (i.e., aspects) should share a similar semantic interpretation, and thus interpretability of a topic can be estimated by measuring how many words are not related. For each aspect  $a_i$  and word  $w$  having been highlighted at least once as belonging to aspect  $a_i$ , we computed the probability  $P(w|a_i)$  on each dataset and sorted them in decreasing order of  $P(w|a_i)$ . Unsurprisingly, we found that the most common words are stop words such as “a” and “it”, because masks are mostly word sequences instead of individual words. To gain a better interpretation of the aspect words, we followed the procedure in McAuley et al. (2012): we first computed the averages across all aspect words for each word  $w$  as follows:

$$b_w = \frac{1}{|A|} \sum_{i=1}^{|A|} P(w|a_i) \quad (\text{C.2})$$

It represents a general distribution that includes words common to all aspects. The final word distribution per aspect is computed by removing the general distribution as follows:

$$\hat{P}(w|a_i) = P(w|a_i) - b_w \quad (\text{C.3})$$

After generating the final word distribution per aspect, we picked the top ten words and asked two human annotators to identify intruder words (i.e., words not matching the corresponding aspect). We show in Table C.1, Table C.2, and also Table C.3 the top ten words for each

## Appendix C. Visualizations

Table C.1: Top ten words for each aspect from the *Beer* dataset, learned by various models. **Red** denotes intruders according to two annotators. Found words are generally noisier due to the high correlation between *Taste* and other aspects. However, *MTM* provides better results than other methods.

	Model	Top-10 Words
Appearance	SAM	<b>nothing</b> beautiful lager nice <b>average</b> macro lagers corn <b>rich</b> gorgeous
	MASA	lacing head lace <b>smell</b> amber retention beer nice carbonation glass
	MAA	head lacing <b>smell aroma</b> color pours amber glass white retention
	MTM (Ours)	head lacing <b>smell</b> white lace retention glass <b>aroma</b> tan thin
Smell	SAM	faint <b>nice mild</b> light slight complex good wonderful grainy great
	MASA	aroma hops nose chocolate caramel malt citrus fruit smell fruits
	MAA	<b>taste</b> hints hint <b>lots t- starts</b> blend mix <b>upfront</b> malts
	MTM (Ours)	<b>taste</b> malt aroma hops sweet citrus caramel nose malts chocolate
Palate	SAM	thin <b>bad</b> light watery creamy silky medium body smooth <b>perfect</b>
	MASA	smooth light medium thin creamy <b>bad</b> watery <b>full</b> crisp <b>clean</b>
	MAA	good beer carbonation smooth <b>drinkable</b> medium bodied <b>nice</b> body <b>overall</b>
	MTM (Ours)	carbonation medium mouthfeel body smooth bodied <b>drinkability</b> creamy light <b>overall</b>
Taste	SAM	<b>decent</b> great complex delicious tasty favorite <b>pretty</b> sweet <b>well best</b>
	MASA	good <b>drinkable nice</b> tasty great enjoyable <b>decent solid</b> balanced <b>average</b>
	MAA	malt hops flavor hop flavors caramel malts bitterness bit chocolate
	MTM (Ours)	malt sweet hops flavor bitterness finish chocolate bitter caramel sweetness

aspect, where **red** denotes all words identified as unrelated to the aspect by the two annotators. Generally, our model finds better sets of words across the three datasets compared with other methods. Additionally, we observe that the aspects can be easily recovered, given its top words.

## C.1 Extracted Topic Words per Aspect from Rationales

Table C.2: Top ten words for each aspect from the *Hotel* dataset, learned by various models. **Red** denotes intruders according to human annotators. Besides SAM, all methods find similar words for most aspects except the aspect *Value*. The top words of *MTM* do not contain any intruder.

	Model	Top-10 Words
Service	SAM	staff service friendly nice told helpful good great lovely manager
	MASA	friendly helpful told rude nice good pleasant asked enjoyed worst
	MAA	staff service helpful friendly nice good rude excellent great desk
	MTM (Ours)	staff friendly service desk helpful manager reception told rude asked
Cleanliness	SAM	clean cleaned dirty toilet smell cleaning sheets comfortable nice hair
	MASA	clean dirty cleaning spotless stains cleaned cleanliness mold filthy bugs
	MAA	clean dirty cleaned filthy stained well spotless carpet sheets stains
	MTM (Ours)	clean dirty bathroom room bed cleaned sheets smell carpet toilet
Value	SAM	good stay great well <b>dirty</b> recommend worth definitely <b>friendly</b> charged
	MASA	great good poor excellent terrible awful <b>dirty</b> horrible <b>disgusting comfortable</b>
	MAA	<b>night stayed</b> stay <b>nights 2 day</b> price <b>water 4 3</b>
	MTM (Ours)	good price expensive paid cheap worth better pay overall disappointed
Location	SAM	location close far place walking <b>definitely</b> located <b>stay</b> short view
	MASA	location beach walk hotel town located restaurants walking close taxi
	MAA	location hotel place located close far area beach view situated
	MTM (Ours)	location great area walk beach hotel town close city street
Room	SAM	<b>dirty clean</b> small best comfortable large worst modern <b>smell</b> spacious
	MASA	comfortable small spacious nice large dated well tiny modern basic
	MAA	room rooms bathroom bed spacious small beds large shower modern
	MTM (Ours)	comfortable room small spacious nice modern rooms large tiny walls

Table C.3: Top ten words for each aspect from the *decorrelated Beer* dataset, learned by various models. **Red** denotes intruders according to two annotators. For the three aspects, MTM has only one word considered as an intruder, followed by MASA with SAM (two) and MAA (six).

	Model	Top-10 Words
Appearance	SAM	head color white brown dark lacing <b>pours</b> amber clear black
	MASA	head lacing lace retention glass foam color amber yellow cloudy
	MAA	nice dark amber <b>pours</b> black hazy brown <b>great</b> cloudy clear
	MTM (Ours)	head color lacing white brown clear amber glass black retention
Smell	SAM	sweet malt hops coffee chocolate citrus hop strong smell aroma
	MASA	smell aroma nose smells sweet aromas scent hops malty roasted
	MAA	<b>taste</b> smell aroma sweet chocolate <b>lacing</b> malt roasted hops nose
	MTM (Ours)	smell aroma nose smells sweet malt citrus chocolate caramel aromas
Palate	SAM	mouthfeel smooth medium carbonation bodied watery body thin creamy <b>full</b>
	MASA	mouthfeel medium smooth body <b>nice</b> m- feel bodied mouth <b>beer</b>
	MAA	carbonation mouthfeel medium <b>overall</b> smooth finish body <b>drinkability</b> bodied watery
	MTM (Ours)	mouthfeel carbonation medium smooth body bodied <b>drinkability</b> good mouth thin

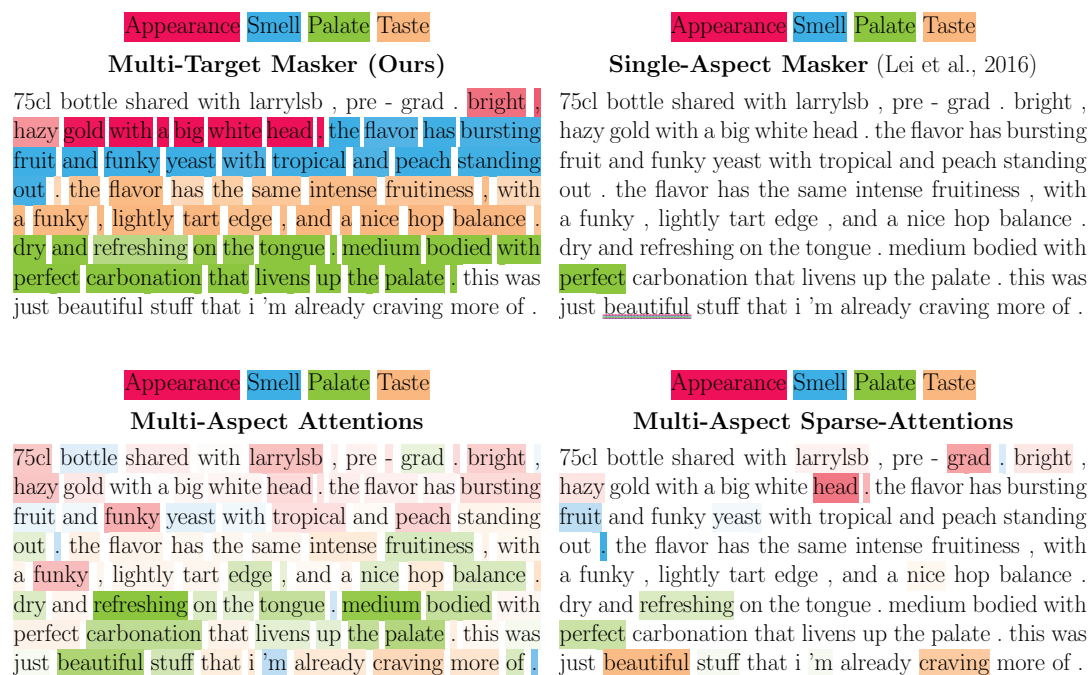


Figure C.1: A sample review from the *Beer* dataset, with computed masks from different methods. *MTM* achieves near-perfect annotations, while *SAM* highlights only two words where one is ambiguous with respect to the four aspects. *MAA* mixes between the aspect *Appearance* and *Smell*. *MASA* identifies some words but lacks coverage.

## C.2 Multi-Dimensional Rationales

We randomly sampled reviews from each dataset and computed the masks and attentions of four models: our Multi-Target Masker (*MTM*), the Single-Aspect Masker (*SAM*) (Lei et al., 2016), and two attention models with additive and sparse attention, called Multi-Aspect Attentions (*MAA*) and Multi-Aspect Sparse-Attentions (*MASA*), respectively (see Section 2.5.3). Each color represents an aspect and the shade its confidence. All models generate soft attentions or masks besides *SAM*, which does hard masking. Samples for the *Beer*, *Hotel*, and *Decorrelated Beer* datasets are shown in Figures C.1, C.2, C.3, C.4, C.5 and C.6 respectively.



Figure C.2: A sample review from the *Beer* dataset, with computed masks from different methods. *MTM* can accurately identify what parts of the review describe each aspect. *MAA* provides very noisy labels due to the high imbalance and correlation between aspects, while *MASA* highlights only a few important words. We can see that *SAM* is confused and performs a poor selection.

## Appendix C. Visualizations



Figure C.3: A sample review from the *Hotel* dataset, with computed masks from different methods. *MTM* emphasizes consecutive words, identifies essential spans while having a small amount of noise. *SAM* focuses on certain specific words and spans, but labels are ambiguous. The *MAA* model highlights many words, ignores a few crucial keyphrases, but labels are noisy when the confidence is low. *MASA* provides noisier tags than *MAA*.





Figure C.4: A sample review from the *Hotel* dataset, with computed masks from different methods. Our *MTM* model finds most of the crucial span of words with a small amount of noise. *SAM* lacks coverage but identifies words where half are correctly tags and the others ambiguous. *MAA* partially correctly highlights words for the aspects *Service*, *Location*, and *Value* while missing out on the aspect *Cleanliness*. *MASA* confidently finds a few important words.

## Appendix C. Visualizations

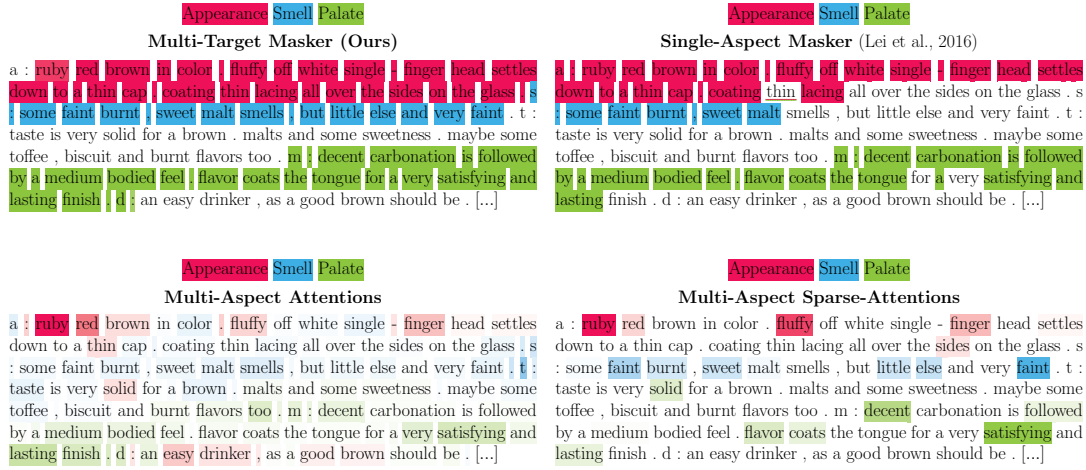


Figure C.5: A sample review from the *decorrelated Beer* dataset, with computed masks from different methods. Our model *MTM* highlights all the words corresponding to the aspects. *SAM* only highlights the most crucial information, but some words are missing out and one is ambiguous. *MAA* and *MASA* fail to identify most of the words related to the aspect *Appearance*, and only a few words have high confidence, resulting in noisy labeling. Additionally, *MAA* considers words belonging to the aspect *Taste* whereas this dataset does not include it in the aspect set (because it has a high correlation with other rating scores).

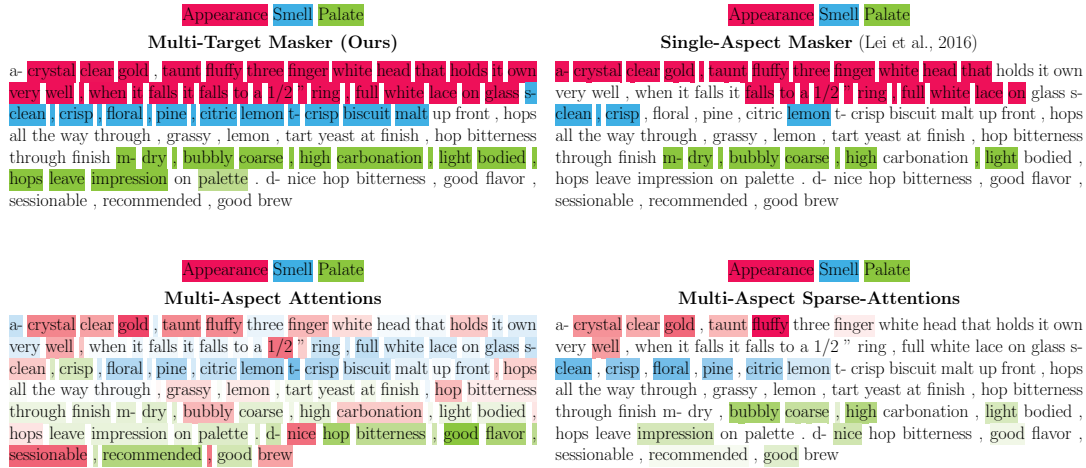


Figure C.6: A sample review from the *decorrelated Beer* dataset, with computed masks from different methods. *MTM* finds the exact parts corresponding to the aspect *Appearance* and *Palate* while covering most of the aspect *Smell*. *SAM* identifies key-information without any ambiguity, but lacks coverage. *MAA* highlights confidently nearly all the words while having some noise for the aspect *Appearance*. *MASA* selects confidently only most predictive words.

Appearance Aroma Palate Taste Overall					
ConRAT (Ours)					InvRAT Chang et al. (2020)
on-tap at lagunitas a : the pour is a hazy straw color with an initially fluffy white head that slowly dies down to a thin layer . s : an amazingly funky and tart aroma is present immediately . lots of sour apples , lemons , and maybe some green grapes along with a subtle wood character and a bit of grass . t : the tartness is bright and green , lots of lemons and apples . the oak , grass , wet straw , and mild earthiness give this beer a great funky balance to the sourness . m : the body is somewhat light and crisp , with a great level of effervescence and slight dryness . d : this is an absolutely fantastic beer . i would drink this like nobody 's business if it was more readily available and/or lagunitas was n't such a drive .					on-tap at lagunitas a : the pour is a hazy straw color with an initially fluffy white head that slowly dies down to a thin layer . s : an amazingly funky and tart aroma is present immediately . lots of sour apples , lemons , and maybe some green grapes along with a subtle wood character and a bit of grass . t : the tartness is bright and green , lots of lemons and apples . the oak , grass , wet straw , and mild earthiness give this beer a great funky balance to the sourness . m : the body is somewhat light and crisp , with a great level of effervescence and slight dryness . d : this is an absolutely fantastic beer . i would drink this like nobody 's business if it was more readily available and/or lagunitas was n't such a drive .
ConRAT (Ours)					RNP Lei et al. (2016)
pours out in a opaque dark yellow colour . topped with a large , thick white foam . very cotton-like but fruity and strong aroma of oranges , peaches and banana with undertones of coriander . it also has some weak vinous accents thick and wheaty flavour of cloves , banana , apricots and oranges . thick , full and round mouthfeel . quite tart in the back of the throat bananas in the long velvety soft finish with hoppy note from orange-peels . a wonderfull winter wheat , too bad it was only 5000 bottles made					pours out in a opaque dark yellow colour . topped with a large , thick white foam . very cotton-like but fruity and strong aroma of oranges , peaches and banana with undertones of coriander . it also has some weak vinous accents thick and wheaty flavour of cloves , banana , apricots and oranges . thick , full and round mouthfeel . quite tart in the back of the throat bananas in the long velvety soft finish with hoppy note from orange-peels . a wonderfull winter wheat , too bad it was only 5000 bottles made

Figure C.7: Examples of generated rationales with  $\ell = 10$  for a beer review. Underline highlights ambiguities.

### C.3 Concepts

Additional samples of generated rationales are shown in Figure C.7 and C.8. We can observe that baselines suffer from spurious correlations: the rationale for the aspect Aroma, Palate, and Taste are often exchanged, or several rationales pick the same text snippets. On the other hand, ConRAT finds better concepts while only trained on the overall aspect label. As it has been shown in prior work (Lei et al., 2016; Chang et al., 2020; Antognini et al., 2021b) rationale methods suffer from the high correlation between rating scores because each model is trained independently for each aspect. Therefore, they rely on the assumption that the data have low internal correlations, which does not reflect the real data distribution. By contrast, ConRAT alleviates this problem by finding all concepts in one training.

## Appendix C. Visualizations

ConRAT (Ours)	InvRAT Chang et al. (2020)	RNP Lei et al. (2016)
a : pours a clear dark amber colour . with a thick two finger creamy off white head . settles to a small cap . leaves quite a bit of lacing . s : caramel malt with a grainy smell . also a bit of a fruity smell closer to dark fruits t : caramel malt up front with a grainy taste . then it finishes with a more sweet dark fruity taste . finishes dry . m : medium carbonation with a medium body d : it 's a decent beer . nothing great but gets the job done if you enjoy this style .	a : pours a clear dark amber colour . with a thick two finger creamy off white head . settles to a small cap . leaves quite a bit of lacing . s : caramel malt with a grainy smell . also a bit of a fruity smell closer to dark fruits t : caramel malt up front with a more grainy taste . then it finishes with a more sweet dark fruity taste . finishes dry . m : medium carbonation with a medium body d : it 's a decent beer . nothing great but it gets the job done if you enjoy this style .	a : pours a clear dark amber colour . with a thick two finger creamy off white head . settles to a small cap . leaves quite a bit of lacing . s : caramel malt with a grainy smell . also a bit of a fruity smell closer to dark fruits t : caramel malt up front with a more grainy taste . then it finishes with a more sweet dark fruity taste . finishes dry . m : medium carbonation with a medium body d : it 's a decent beer . nothing great but it gets the job done if you enjoy this style .
ConRAT (Ours)	InvRAT Chang et al. (2020)	RNP Lei et al. (2016)
beer review 100 a - pours a light somewhat hazy gold color into my pint glass with about one finger of head moderate retention and very nice lacing . s - strong aroma of hops , pine and grapefruit citrus notes as well as sweet malts . t - to me , this is a great tasting ipa . sweet malts , followed by a very nice pine and citrus hop fusion that finishes with just the right amount of bitterness m - medium in body , crisp and refreshing . d - this drinks great as an ipa , and all you hopheads out there like myself remember this is an ipa , not a double or imperial , and for the category it 's in it is an awesome beer	beer review 100 a - pours a light somewhat hazy gold color into my pint glass with about one finger of head moderate retention and very nice lacing . s - strong aroma of hops , pine and grapefruit citrus notes as well as sweet malts . t - to me , this is a great tasting ipa . sweet malts , followed by a very nice pine and citrus hop fusion that finishes with just the right amount of bitterness m - medium in body , crisp and refreshing . d - this drinks great as an ipa , and all you hopheads out there like myself remember this is an ipa , not a double or imperial , and for the category it 's in it is an awesome beer	beer review 100 a - pours a light somewhat hazy gold color into my pint glass with about one finger of head moderate retention and very nice lacing . s - strong aroma of hops , pine and grapefruit citrus notes as well as sweet malts . t - to me , this is a great tasting ipa . sweet malts , followed by a very nice pine and citrus hop fusion that finishes with just the right amount of bitterness m - medium in body , crisp and refreshing . d - this drinks great as an ipa , and all you hopheads out there like myself remember this is an ipa , not a double or imperial , and for the category it 's in it is an awesome beer

Figure C.8: Examples of generated rationales with  $\ell = 20$  for a beer review. Underline highlights ambiguities.

## D Human Evaluation Details

We use Amazon’s Mechanical Turk crowdsourcing platform to recruit human annotators to evaluate the quality of extracted justifications and the generated justifications produced by each model. To ensure high-quality of the collected data, we restricted the pool to native English speakers from the U.S., U.K., Canada, or Australia. Additionally, we set the worker requirements at a 98% approval rate and more than 1,000 HITS.

### D.1 Understandability of Concepts

The user interface used to judge the quality of the justifications extracted from different methods, in Section 3.5.5, is shown in Figure D.1.

### D.2 Justification Evaluation

The user interface used to judge the quality of the justifications extracted from different methods, in Section 4.6.2, is shown in Figure D.2. Another human assessment evaluates the generated justifications (see Section 4.6.3) on the four dimensions:

1. **Overall** measures the overall subjective quality;
2. **Fluency** represents the readability;
3. **Informativeness** indicates whether the justification contains information pertinent to the user;
4. **Relevance** measures how relevant the information is to an item.

The interface is available in Figure D.3

Appendix D. Human Evaluation Details

Instructions

Please read the following partially-blocked review on a beer and:  
1) guess the aspect that the revealed part is talking about;  
2) guess the score on the aspect that you selected.

.....  
..... the  
malt and hops played well together , with the bitter finish lasting quite awhile . had  
this on tap .....

Select an option

Appearance/Look - Positive	1
Appearance/Look - Negative	2
Smell/Aroma - Positive	3
Smell/Aroma - Negative	4
Palate/Mouthfeel - Positive	5
Palate/Mouthfeel - Negative	6
Taste - Positive	7
Taste - Negative	8
Overall - Positive	9
Overall - Negative	0

Submit

Figure D.1: Annotation platform for judging the quality of the concepts in the subjective evaluation on beer reviews.

## D.2 Justification Evaluation

Please find below an original hotel review with its rating and three automatically generated excerpts. **Which excerpts best/worst justified the overall rating?**

---

**Overall Rating:** 4.0 / 5.0

**Review:** what a lovely tennis resort nestled into the mountain at the end of baristo rd. the location is amazing as it is within walking distance of shops, activities and restaurants of palm springs, the spa casino is also within walking distance also. the desert and mountain views are spectacular, the 3 pools and spas usually not busy and spencer's restaurant is on site. the staff are courteous and eager to please.

---

Excerpt from Model 1	Excerpt from Model 2	Excerpt from Model 3
<ul style="list-style-type: none"><li>the location is amazing</li></ul>	<ul style="list-style-type: none"><li>what a lovely tennis resort</li><li>the location is amazing</li><li>the desert and mountain views are spectacular</li><li>the staff are courteous and eager to please</li></ul>	<ul style="list-style-type: none"><li>the staff are courteous and eager to please</li><li>what a lovely tennis resort nestled into the mountain at the end of baristo rd</li><li>the desert and mountain views are spectacular , the 3 pools and spas usually not busy and spencer 's restaurant is on site</li><li>the location is amazing as it is within walking distance of shops , activities and restaurants of palm springs , the spa casino is also within walking distance also</li></ul>

Please choose the excerpt that best/worst justified the overall rating.

Best: ☐ Model 1 ☐ Model 2 ☒ Model 3

Worst: ☐ Model 1 ☐ Model 2 ☐ Model 3

Figure D.2: Annotation platform for judging the quality of extracted justifications from different methods. The justifications are shown in random order for each comparison. In this example, our method corresponds to the third model.



## Appendix D. Human Evaluation Details

We currently have a system which recommends hotels to users. In addition to the recommendation itself, our system generates a personalized justification of his/her potential overall experience.  
Please find below an original hotel review and four automatically generated justifications.

- Which one of these is the best/least convincing justification while recommending a hotel?
- Which is the best/worst in terms of Fluency, Informativeness, and Relevance?

**Original Review:** Stayed here overnight for the 4th of July weekend. The hotel does a great job keeping the rustic lodge theme going throughout the property. Everything is updated and clean. The standard rooms are a little tight but there were four of us so that may have played a role. They have a very nice indoor pool with plenty of space and a hot tub. The pool area has the lodge décor as well including some artificial rock formations in and around the pool. Good continental breakfast in the morning.

Excerpt from Model 1	Excerpt from Model 2	Excerpt from Model 3	Excerpt from Model 4
this property is located in a nice setting . the property is clean and well maintained . the pool and hot tub are very nice . the breakfast was good and the staff was friendly . the pool was nice and clean . the hot tub was a nice touch . the pool was nice and clean .	<ul style="list-style-type: none"> <li>the pool and hot tub are very nice and there is plenty of space</li> <li>the breakfast spread was good</li> </ul>	<ul style="list-style-type: none"> <li>the pool is nice with plenty of hot tub and space</li> <li>breakfast was good</li> </ul>	the staff was very friendly and helpful

**Overall:**

Which one of the justification is the best/least convincing?

Best: ☐ Model 1 ☐ Model 2 ☒ Model 3 ☐ Model 4

Worst: ☐ Model 1 ☐ Model 2 ☒ Model 3 ☐ Model 4

**Fluency:**

The justification is grammatically correct, easy to read and to understand.

Best: ☐ Model 1 ☐ Model 2 ☒ Model 3 ☐ Model 4

Worst: ☐ Model 1 ☐ Model 2 ☒ Model 3 ☐ Model 4

**Informativeness:**

How helpful/pertinent is the information in the justification to a user?

Best: ☐ Model 1 ☐ Model 2 ☒ Model 3 ☐ Model 4

Worst: ☐ Model 1 ☐ Model 2 ☒ Model 3 ☐ Model 4

**Relevance:**

How relevant is the information in the justification to a hotel?

Best: ☐ Model 1 ☐ Model 2 ☒ Model 3 ☐ Model 4

Worst: ☐ Model 1 ☐ Model 2 ☒ Model 3 ☐ Model 4

**Submit**

Figure D.3: Annotation platform for judging the quality of generated justifications from different methods, on four dimensions. The justifications are shown in random order for each comparison. In this example, our method corresponds to the second model.



## E M&Ms-VAE Derivation

We provide the full derivation of the M&Ms-VAE's evidence lower bound of Equation 5.5.

$$\log p(\mathbf{r}_u, \mathbf{k}_u) = \log \int_{\mathbf{z}_u} p_{\Theta}(\mathbf{r}_u, \mathbf{k}_u, \mathbf{z}_u) d\mathbf{z}_u \quad (\text{E.1})$$

$$= \log \int_{\mathbf{z}_u} p_{\Theta}(\mathbf{r}_u, \mathbf{k}_u, \mathbf{z}_u) \frac{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)}{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} d\mathbf{z}_u \quad (\text{E.2})$$

$$= \log \left( \mathbb{E}_{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} \left[ \frac{p_{\Theta}(\mathbf{r}_u, \mathbf{k}_u, \mathbf{z}_u)}{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} \right] \right) \quad (\text{E.3})$$

$$\geq \mathbb{E}_{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} [\log p_{\Theta}(\mathbf{r}_u, \mathbf{k}_u, \mathbf{z}_u)] - \mathbb{E}_{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} [\log q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)] \quad (\text{E.4})$$

$$= \mathbb{E}_{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} [\log p_{\Theta}(\mathbf{r}_u, \mathbf{k}_u | \mathbf{z}_u)] + \mathbb{E}_{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} [\log p(\mathbf{z}_u)] - \mathbb{E}_{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} [\log q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)] \quad (\text{E.5})$$

$$= \mathbb{E}_{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} [\log p_{\Theta}(\mathbf{r}_u, \mathbf{k}_u | \mathbf{z}_u)] - \text{D}_{\text{KL}}[q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u) \parallel p(\mathbf{z}_u)] \quad (\text{E.6})$$

$$= \mathbb{E}_{q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u)} [\log p_{\Theta_r}(\mathbf{r}_u | \mathbf{z}_u) + \log p_{\Theta_k}(\mathbf{k}_u | \mathbf{z}_u)] - \text{D}_{\text{KL}}[q_{\Phi}(\mathbf{z}_u | \mathbf{r}_u, \mathbf{k}_u) \parallel p(\mathbf{z}_u)] \quad (\text{E.7})$$



# F Reproducibility

## F.1 Multi-Target Masker (MTM)

Most of the time, the model converges under 20 epochs (maximum of 20 and 3 minutes per epoch for the *Beer* and *Hotel* dataset, respectively). The range of hyperparameters are the following for MTM (similar for other models).

- Learning rate: [0.001, 0.0005, 0.00075];
- Hidden size: [50, 100, 200];
- Filter numbers (CNN): [50, 100, 200];
- Bi-directional (LSTM): [*True*, *False*];
- Dropout: [0, 0.1, 0.2];
- Weight decay: [0,  $1e^{-6}$ ,  $1e^{-8}$ ,  $1e^{-10}$ ];
- Gumbel Temperature  $\tau$  : [0.5, 0.8, 1.0, 1.2];
- $\lambda_{sel}$ : [0.01, 0.02, 0.03, 0.04, 0.05];
- $\lambda_p$ : [0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15];
- $\lambda_{cont}$ : [0.02, 0.04, 0.06, 0.08, 0.10];

### Hardware / Software

- **CPU**: 2x Intel Xeon E5-2680 v3, 2x 12 cores, 24 threads, 2.5 GHz, 30 MB cache;
- **RAM**: 16x16GB DDR4-2133;
- **GPU**: 1x Nvidia Titan X;

- **OS:** Ubuntu 18.04;
- **Software:** Python 3.6, PyTorch 1.3.1, CUDA 9.2.

### F.2 Concept-based Rationalizer (ConRAT)

We tune all models on the dev set. We truncate all reviews to 320 tokens for the beer dataset and 400 tokens for Amazon reviews. We have operated a random search over 16 trials. All baselines, except CAR, are tuned for each aspect (80 trials in total for the five aspects). We chose the models achieving the lowest validation accuracy. Most of the time, all models converged under 30 epochs. The range of hyperparameters are the following for ConRAT (similar for other models):

- Learning rate: [0.0005, 0.00075, 0.001];
- Batch size: [128];
- Hidden size: [256];
- $\lambda_D$ : [0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0];
- $\lambda_O$ : [0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0];
- $\lambda_T$ : [0.5, 0.6];
- Dropout: [0.0, 0.1];
- Weight decay: [0.0,  $10^{-8}$ ,  $10^{-10}$ ];
- Gumbel temperature in  $f_\theta(\cdot)$ : [1.0; 1.5];
- Gumbel temperature in  $s_\theta(\cdot)$ : [1.0; 1.5];

#### Hardware / Software

- **CPU:** 2x Intel Xeon E5-2680 v3, 2x 12 cores, 24 threads, 2.5 GHz, 30 MB cache;
- **RAM:** 16x16GB DDR4-2133;
- **GPU:** 2x Nvidia Titan X Maxwell;
- **OS:** Ubuntu 18.04;
- **Software:** Python 3.6, PyTorch 1.3, CUDA 10.

### F.3 A Multi-Task Transformer with Explanations and Critiquing (T-RECS)

We build the justification history  $J^u, J^i$ , with  $N_{just} = 32$ . In T-RECS, we set the embedding, latent, and self-attention dimension size to 256, and the dimension of the feed-forward network to 1024. The encoder and decoder consist of two layers of Transformer with 4 attention heads. We use a batch size of 128, dropout of 0.1, 4000 warm-up steps, smoothing parameter  $\varepsilon = 0.1$ , and Adam with learning rate 0.001,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$ , and  $\epsilon = 10^{-9}$ . The Rating Classifier and Keyphrase Explainer are two layers of 128 and 64 dimensions with LeakyReLU ( $\alpha = 0.2$ ). For critiquing, we choose a threshold and decay coefficient  $T = 0.015, \zeta = 0.9$  and  $T = 0.01, \zeta = 0.975$  for hotel and beer reviews, respectively. We use the code from the authors for most models. We tune all models on the dev set. We have operated a random search over 10 trials. We chose the models achieving the lowest validation loss. The range of hyperparameters are the following for T-RECS (similar for other models):

- Learning rate: [0.001, 0.0001];
- Max epochs: [100, 200, 300];
- Batch size: [128];
- Hidden size encoder/decoder: [256];
- Attention heads: [4];
- Number of layers: [2];
- Dropout encoder: [0.0, 0.1, 0.2, 0.3, 0.4, 0.5];
- Dropout decoder: [0.0, 0.1, 0.2, 0.3];
- General dropout: [0.0, 0.1, 0.2];
- Warmup: [2000, 4000, 8000, 16000];
- $\lambda_r, \lambda_{kp}, \lambda_{just}$ : [1.0];

Most of the time, the model converges under 20 epochs. For critiquing, we employed:

- Decay coefficient  $\zeta$ : [0.5, 0.75, 0.8, 0.9, 0.95];
- Max iterations: [25, 50, 75, 100, 200];
- Threshold: [0.015, 0.01, 0.005];

### Hardware / Software

- **CPU:** 2x Intel Xeon E5-2680 v3 (Haswell), 2x 12 cores, 24 threads, 2.5 GHz, 30 MB cache;
- **RAM:** 16x 16GB DDR4-2133;
- **GPU:** 2x Nvidia Titan X Maxwell;
- **OS:** Ubuntu 18.04;
- **Software:** Python 3.6, PyTorch 1.3.0, CUDA 10.0.

### F.4 MultiModal Variational AutoEncoder (M&Ms-VAE)

The official baselines' codes from the respective authors, including the tuning procedure, are available in <sup>1234</sup>. The final hyperparameters for all models and datasets are shown in Table F.1. For all experiments, we used the following hardware:

### Hardware / Software

- **CPU:** 2x Intel Xeon E5-2680 v3, 2x 12 cores, 24 threads, 2.5 GHz, 30 MB cache;
- **RAM:** 16x16GB DDR4-2133;
- **GPU:** 1x Nvidia Titan X Maxwell;
- **OS:** Ubuntu 18.04;
- **Software:** Python 3.6, PyTorch 1.6.1, CUDA 10.2.

---

<sup>1</sup>[https://github.com/wuga214/NCE\\_Projected\\_LRec](https://github.com/wuga214/NCE_Projected_LRec)

<sup>2</sup><https://github.com/k9luo/DeepCritiquingForVAEBasedRecSys>

<sup>3</sup><https://github.com/wuga214/DeepCritiquingForRecSys>

<sup>4</sup><https://github.com/litosly/RankingOptimizationApproachtoLLC>

#### F.4 MultiModal Variational AutoEncoder (M&Ms-VAE)

Table F.1: Best hyperparameter setting for each model. The top table refers to Section 5.5.3 and the bottom one to Section 5.5.4.

Dataset	Model	$H$	$LR$	$\lambda_{L2}$	$\lambda$	$\lambda_{KP}$	$\lambda_C$	$\beta$	Iteration	Epoch	Dropout	$\gamma$	Neg. Samples
<i>Beer</i>	AutoRec	200	0.0001	0.00001	1.0	-	-	-	-	300	-	-	-
	BPR	200	-	0.0001	1.0	-	-	-	-	30	-	-	1
	CDAE	200	0.0001	0.00001	1.0	-	-	-	-	300	0.2	-	-
	NCE-PLRec	50	-	10000.0	1.0	-	-	-	10	-	-	1.1	-
	PLRec	400	-	10000.0	1.0	-	-	-	10	-	-	-	-
	PureSVD	50	-	-	-	-	-	-	10	-	-	-	-
	VAE-CF	50	0.0001	0.0001	1.0	-	-	0.2	-	-	0.4	-	-
	CE-VAE	100	0.0001	0.0001	1.0	0.01	0.01	0.001	-	300	0.5	-	-
	CE-VNCF	100	0.0005	0.00005	1.0	1.0	1.0	0.1	-	100	0.1	-	5
	M&Ms-VAE	300	0.00005	1e-10	3.0	-	-	0.7	-	300	0.4	-	-
<i>CDs&amp;Vinyl</i>	AutoRec	200	0.0001	0.00001	1.0	-	-	-	-	300	-	-	-
	BPR	200	-	0.0001	1.0	-	-	-	-	30	-	-	1
	CDAE	200	0.0001	0.00001	1.0	-	-	-	-	300	0.2	-	-
	NCE-PLRec	200	-	1000.0	1.0	-	-	-	10	-	-	1.3	-
	PLRec	400	-	1000.0	1.0	-	-	-	10	-	-	-	-
	PureSVD	200	-	-	-	-	-	10	-	-	-	-	-
	VAE-CF	200	0.0001	0.00001	1.0	-	-	0.2	-	-	0.3	-	-
	CE-VAE	200	0.0001	0.0001	1.0	0.001	0.001	0.0001	-	600	0.5	-	-
	CE-VNCF	100	0.0001	0.0001	1.0	1.0	1.0	0.1	-	100	0.1	-	5
	M&Ms-VAE	400	0.00005	1e-12	1.0	-	-	0.4	-	400	0.4	-	-
<i>Yelp</i>	AutoRec	50	0.0001	0.001	1.0	-	-	-	-	300	-	-	-
	BPR	100	-	0.0001	1.0	-	-	-	-	30	-	-	1
	CDAE	50	0.0001	0.001	1.0	-	-	-	-	300	0.4	-	-
	NCE-PLRec	50	-	10000.0	1.0	-	-	-	10	-	-	1.3	-
	PLRec	400	-	10000.0	1.0	-	-	-	10	-	-	-	-
	PureSVD	50	-	-	-	-	-	-	10	-	-	-	-
	VAE-CF	50	0.0001	0.001	1.0	-	-	0.2	-	-	0.2	-	-
	CE-VAE	200	0.0001	0.0001	1.0	0.01	0.01	0.001	-	600	0.4	-	-
	CE-VNCF	100	0.0005	0.0001	1.0	1.0	1.0	0.1	-	100	0.1	-	5
	M&Ms-VAE	500	0.00005	1e-10	10.0	-	-	0.8	-	300	0.7	-	-
<i>Hotel</i>	AutoRec	50	0.0001	1e-05	1.0	-	-	-	-	300	-	-	-
	BPR	200	-	0.0001	1.0	-	-	-	-	30	-	-	1
	CDAE	200	0.0001	0.001	1.0	-	-	-	-	300	0.2	-	-
	NCE-PLRec	50	-	10000.0	1.0	-	-	-	10	-	-	1.3	-
	PLRec	400	-	10000.0	1.0	-	-	-	10	-	-	-	-
	PureSVD	50	-	-	-	-	-	-	10	-	-	-	-
	VAE-CF	50	0.0001	1e-05	1.0	-	-	0.2	-	-	0.5	-	-
	CE-VAE	200	0.0001	0.0001	1.0	0.01	0.01	0.001	-	600	0.2	-	-
	CE-VNCF	100	0.0005	0.0001	1.0	1.0	1.0	0.1	-	100	0.1	-	5
	M&Ms-VAE	400	0.00005	1e-12	2.0	-	-	0.8	-	300	-	-	-

Dataset	Model	$h$	$LR$	$\lambda_{L2}$
Beer		0.75	0.001	0
CDsVinyl	M&Ms-VAE $\xi(\cdot)$	3.0	0.001	1e-10
Yelp	(Critiquing)	2.0	0.001	0
Hotel		5.0	0.001	1e-10





# Bibliography

- Alaniz, S. and Akata, Z. (2019). Explainable observer-classifier for explainable binary decisions. *arXiv preprint arXiv:1902.01780*.
- Aletras, N. and Stevenson, M. (2013). Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 13–22, Potsdam, Germany. Association for Computational Linguistics.
- Alvarez-Melis, D. and Jaakkola, T. (2017). A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421, Copenhagen, Denmark. Association for Computational Linguistics.
- Alvarez-Melis, D. and Jaakkola, T. S. (2018). Towards robust interpretability with self-explaining neural networks. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7786–7795.
- Amershi, S., Cakmak, M., Knox, W. B., and Kulesza, T. (2014). Power to the people: The role of humans in interactive machine learning. volume 35, pages 105–120.
- Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine bias. *ProPublica*, May, 23:2016.
- Antognini, D. and Faltings, B. (2019). Learning to create sentence semantic relation graphs for multi-document summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 32–41, Hong Kong, China. Association for Computational Linguistics.
- Antognini, D. and Faltings, B. (2020a). GameWikiSum: a novel large multi-document summarization dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6645–6650, Marseille, France. European Language Resources Association.
- Antognini, D. and Faltings, B. (2020b). HotelRec: a novel very large-scale hotel recommendation dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4917–4923, Marseille, France. European Language Resources Association.

## Bibliography

---

- Antognini, D. and Faltings, B. (2021a). Fast multi-step critiquing for vae-based recommender systems. In *Fifteenth ACM Conference on Recommender Systems, RecSys '21*, page 209–219, New York, NY, USA. Association for Computing Machinery.
- Antognini, D. and Faltings, B. (2021b). Rationalization through concepts. In *Findings of the Association for Computational Linguistics: ACL 2021*, pages 761–775, Online. Association for Computational Linguistics.
- Antognini, D. and Faltings, B. (2022). Positive & negative critiquing for vae-based recommenders. *arXiv preprint arXiv:2204.02162*.
- Antognini, D., Li, S., Faltings, B., and McAuley, J. (2023). Assistive recipe editing through critiquing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online. Association for Computational Linguistics.
- Antognini, D., Musat, C., and Faltings, B. (2021a). Interacting with explanations through critiquing. In Zhou, Z.-H., editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 515–521. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Antognini, D., Musat, C., and Faltings, B. (2021b). Multi-dimensional explanation of target variables from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence, (AAAI 2021)*, volume 35, pages 12507–12515.
- Armstrong, S. L., Gleitman, L. R., and Gleitman, H. (1983). What some concepts might not be. *Cognition*, 13(3):263–308.
- Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.
- Balog, K., Radlinski, F., and Karatzoglou, A. (2021). *On Interpretation and Measurement of Soft Attributes for Recommendation*, page 890–899. Association for Computing Machinery, New York, NY, USA.
- Bao, Y., Chang, S., Yu, M., and Barzilay, R. (2018). Deriving machine attention from human rationales. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1903–1913, Brussels, Belgium.
- Barocas, S., Hardt, M., and Narayanan, A. (2019). *Fairness and Machine Learning*. fairml-book.org. <http://www.fairmlbook.org>.
- Bastings, J., Aziz, W., and Titov, I. (2019). Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

- Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549.
- Bellini, V., Schiavone, A., Di Noia, T., Ragone, A., and Di Sciascio, E. (2018). Knowledge-aware autoencoders for explainable recommender systems. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*.
- Bengio, Y., Léonard, N., and Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.
- Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., and Jaggi, M. (2018). Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium. Association for Computational Linguistics.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.
- Bouchacourt, D. and Denoyer, L. (2019). Educe: Explaining model decisions through unsupervised concepts extraction. *arXiv preprint arXiv:1905.11852*.
- Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. In Chiarcos, C., de Castilho, E., and Stede, M., editors, *Von der Form zur Bedeutung: Texte automatisch verarbeiten / From Form to Meaning: Processing Texts Automatically, Proceedings of the Biennial GSCL Conference 2009*, pages 31–40, Tübingen. Gunter Narr Verlag.
- Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.
- Brand, A., Allen, L., Altman, M., Hlava, M., and Scott, J. (2015). Beyond authorship: attribution, contribution, collaboration, and credit. *Learned Publishing*, 28(2):151–155.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Buolamwini, J. and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In Friedler, S. A. and Wilson, C., editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91. PMLR.

## Bibliography

---

- Burke, R. D., Hammond, K. J., and Young, B. C. (1996). Knowledge-based navigation of complex information spaces. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, pages 462–468. AAAI Press.
- Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1721–1730, New York, NY, USA. Association for Computing Machinery.
- Castelo, N., Bos, M. W., and Lehmann, D. R. (2019). Task-dependent algorithm aversion. *Journal of Marketing Research*, 56(5):809–825.
- Catherine, R. and Cohen, W. (2017). Transnets: Learning to transform for recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 288–296, New York, NY, USA. Association for Computing Machinery.
- Chang, S., Harper, F. M., and Terveen, L. G. (2016). Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 175–182, New York, NY, USA. Association for Computing Machinery.
- Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. (2019). A game theoretic approach to class-wise selective rationalization. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. (2020). Invariant rationalization. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1448–1458. PMLR.
- Chen, C., Zhang, M., Liu, Y., and Ma, S. (2018a). Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 1583–1592. International World Wide Web Conferences Steering Committee.
- Chen, J., Song, L., Wainwright, M., and Jordan, M. (2018b). Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 883–892.
- Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. (2019a). L-shapley and c-shapley: Efficient model interpretation for structured data. In *International Conference on Learning Representations*.
- Chen, L. and Pu, P. (2007a). Hybrid critiquing-based recommender systems. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, IUI '07, page 22–31, New York, NY, USA. Association for Computing Machinery.

- Chen, L. and Pu, P. (2007b). Preference-based organization interfaces: Aiding user critiques in recommender systems. In *Proceedings of the 11th International Conference on User Modeling*, UM '07, page 77–86, Berlin, Heidelberg. Springer-Verlag.
- Chen, L. and Pu, P. (2012). Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22(1-2).
- Chen, Z., Wang, X., Xie, X., Parsana, M., Soni, A., Ao, X., and Chen, E. (2020). Towards explainable conversational recommendation. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2994–3000. International Joint Conferences on Artificial Intelligence Organization. Main track.
- Chen, Z., Wang, X., Xie, X., Wu, T., Bu, G., Wang, Y., and Chen, E. (2019b). Co-attentive multi-task learning for explainable recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2137–2143. International Joint Conferences on Artificial Intelligence Organization.
- Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Citron, D. K. and Pasquale, F. (2014). The scored society: Due process for automated predictions. *Wash. L. Rev.*, 89:1.
- Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does BERT look at? an analysis of BERT’s attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy. Association for Computational Linguistics.
- Costa, F., Ouyang, S., Dolog, P., and Lawlor, A. (2018). Automatic generation of natural language explanations. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion, IUI '18 Companion*, New York, NY, USA. Association for Computing Machinery.
- Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10*, pages 39–46, New York, NY, USA. Association for Computing Machinery.
- Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. (2020). Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations (ICLR)*.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R., and Wallace, B. C. (2020). ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th*

## Bibliography

---

- Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.
- Dieng, A. B., Wang, C., Gao, J., and Paisley, J. W. (2017). Topicrnn: A recurrent neural network with long-range semantic dependency. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Dietvorst, B. J., Simmons, J. P., and Massey, C. (2015). Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 144(1):114.
- Dietvorst, B. J., Simmons, J. P., and Massey, C. (2018). Overcoming algorithm aversion: People will use imperfect algorithms if they can even slightly modify them. *Manage. Sci.*, 64(3):1155–1170.
- Dong, L., Huang, S., Wei, F., Lapata, M., Zhou, M., and Xu, K. (2017). Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 623–632, Valencia, Spain. Association for Computational Linguistics.
- Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.
- Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of AI research*, 22:457–479.
- Faltings, B., Pu, P., Torrens, M., and Viappiani, P. (2004a). Designing example-critiquing interaction. In *Proceedings of the 9th International Conference on Intelligent User Interfaces, IUI '04*, pages 22–29, Island of Madeira, Portugal. ACM Press.
- Faltings, B., Torrens, M., and Pu, P. (2004b). Solution generation with qualitative models of preferences. *Computational Intelligence*, 20(2).
- Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015a). Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.
- Faruqui, M., Tsvetkov, Y., Yogatama, D., Dyer, C., and Smith, N. A. (2015b). Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1491–1500, Beijing, China. Association for Computational Linguistics.
- Figurnov, M., Mohamed, S., and Mnih, A. (2018). Implicit reparameterization gradients. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R.,

- editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Filipovic, M., Mitrevski, B., Antognini, D., Lejal Glaude, E., Faltings, B., and Musat, C. (2021). Modeling online behavior in recommender systems: The importance of temporal context. In *Proceedings of the Perspectives on the Evaluation of Recommender Systems Workshop at RecSys 2021 (PERSPECTIVE 2021)*.
- Ghorbani, A., Wexler, J., Zou, J. Y., and Kim, B. (2019). Towards automatic concept-based explanations. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Giannakopoulos, A., Antognini, D., Musat, C., Hossmann, A., and Baeriswyl, M. (2017). Dataset construction via attention for aspect term extraction with distant supervision. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 373–380. IEEE.
- Goodman, B. and Flaxman, S. (2017). European union regulations on algorithmic decision-making and a “right to explanation”. *AI Magazine*, 38(3):50–57.
- Goyal, Y., Feder, A., Shalit, U., and Kim, B. (2019). Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*.
- Gunning, D. (2016). *Explainable artificial intelligence (XAI): Technical Report*. Defense Advanced Research Projects Agency DARPA-BAA-16-53, DARPA, Arlington, USA.
- He, R. and McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW '16*, pages 507–517, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW '17*, page 173–182, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Herbelot, A. and Vecchi, E. M. (2015). Building a shared world: mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Lisbon, Portugal. Association for Computational Linguistics.
- Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, CSCW '00*, page 241–250, New York, NY, USA. Association for Computing Machinery.
- Hidasi, B. and Karatzoglou, A. (2018). Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on*

## Bibliography

---

- Information and Knowledge Management*, CIKM '18, pages 843–852, New York, NY, USA. Association for Computing Machinery.
- Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2016). Session-based recommendations with recurrent neural networks. In *the International Conference on Learning Representations (ICLR)*.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *the International Conference on Learning Representations (ICLR)*.
- Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). The curious case of neural text degeneration. In *International Conference on Learning Representations*.
- Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469.
- Jacovi, A. and Goldberg, Y. (2020). Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.
- Jain, S. and Wallace, B. C. (2019). Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jain, S., Wiegrefe, S., Pinter, Y., and Wallace, B. C. (2020). Learning to faithfully rationalize by construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473, Online. Association for Computational Linguistics.
- Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26*.
- Jannach, D. and Jugovac, M. (2019). Measuring the business value of recommender systems. *ACM Trans. Manage. Inf. Syst.*, 10(4).
- Jussupow, E., Benbasat, I., and Heinzl, A. (2020). Why are we averse towards algorithms? a comprehensive literature review on algorithm aversion. In Rowe, F, editor, *28th European Conference on Information Systems - Liberty, Equality, and Fraternity in a Digitizing World*,



- ECIS 2020, Marrakech, Morocco, June 15-17, 2020 : Proceedings*, page RP 168, Atlanta, GA. AISel.
- Karpathy, A., Johnson, J., and Fei-Fei, L. (2015). Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.
- Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.
- Kim, B., Shah, J. A., and Doshi-Velez, F. (2015). Mind the gap: A generative approach to interpretable feature selection and extraction. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Kim, B., Wattenberg, M., Gilmer, J., Cai, C. J., Wexler, J., Viégas, F. B., and Sayres, R. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In Dy, J. G. and Krause, A., editors, *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 2673–2682. PMLR.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.
- Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Kiritchenko, S. and Mohammad, S. M. (2016). Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 811–817, San Diego, California. Association for Computational Linguistics.
- Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., and Newell, C. (2012). Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4):441–504.
- Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. (2020). Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*.
- Konstan, J. A. and Riedl, J. (2012). Recommender systems: From algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22(1–2):101–123.
- Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. (2019). Revealing the dark secrets of BERT. 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)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

## Bibliography

---

- Kunkel, J., Donkers, T., Barbu, C.-M., and Ziegler, J. (2018). Trust-related effects of expertise and similarity cues in human-generated recommendations. *Companion Proceedings of the 23rd International on Intelligent User Interfaces: 2nd Workshop on Theory-Informed User Modeling for Tailoring and Personalizing Interfaces (HUMANIZE)*.
- Kunkel, J., Donkers, T., Michael, L., Barbu, C.-M., and Ziegler, J. (2019). Let me explain: Impact of personal and impersonal explanations on trust in recommender systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 1–12, New York, NY, USA. Association for Computing Machinery.
- Lau, J. H. and Baldwin, T. (2016). The sensitivity of topic coherence evaluation to topic cardinality. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–487.
- Lau, J. H., Newman, D., and Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539, Gothenburg, Sweden. Association for Computational Linguistics.
- Lei, T., Barzilay, R., and Jaakkola, T. (2016). Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.
- Li, H., Sanner, S., Luo, K., and Wu, G. (2020). A ranking optimization approach to latent linear critiquing for conversational recommender systems. In *Fourteenth ACM Conference on Recommender Systems, RecSys '20*, pages 13–22, New York, NY, USA. Association for Computing Machinery.
- Li, J., Chen, X., Hovy, E., and Jurafsky, D. (2016a). Visualizing and understanding neural models in nlp. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691.
- Li, J., Monroe, W., and Jurafsky, D. (2016b). Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.
- Li, J., Yang, H., and Zong, C. (2018a). Document-level multi-aspect sentiment classification by jointly modeling users, aspects, and overall ratings. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 925–936, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Li, O., Liu, H., Chen, C., and Rudin, C. (2018b). Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Li, P. and Tuzhilin, A. (2019). Towards controllable and personalized review generation. 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), pages 3237–3245, Hong Kong, China. Association for Computational Linguistics.
- Li, P., Wang, Z., Bing, L., and Lam, W. (2019). Persona-aware tips generation? In *The World Wide Web Conference, WWW '19*, page 1006–1016, New York, NY, USA. Association for Computing Machinery.
- Li, P., Wang, Z., Ren, Z., Bing, L., and Lam, W. (2017). Neural rating regression with abstractive tips generation for recommendation. In *the ACM SIGIR conference on Research and Development in Information Retrieval*.
- Liang, D., Krishnan, R. G., Hoffman, M. D., and Jebara, T. (2018). Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference, WWW '18*, page 689–698, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Lin, C.-Y. and Hovy, E. (2002). Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*, pages 45–51. Association for Computational Linguistics.
- Lin, Z., Feng, M., dos Santos, C. N., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Linden, G., Hanks, S., and Lesh, N. (1997). Interactive assessment of user preference models: The automated travel assistant. In Jameson, A., Paris, C., and Tasso, C., editors, *User Modeling*, pages 67–78, Vienna. Springer Vienna.
- Louis, A., Roth, D., and Radlinski, F. (2020). “I’d rather just go to bed”: Understanding indirect answers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7411–7425, Online. Association for Computational Linguistics.
- Louviere, J. J., Flynn, T. N., and Marley, A. A. J. (2015). *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press.
- Lu, Y., Dong, R., and Smyth, B. (2018). Why i like it: Multi-task learning for recommendation and explanation. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18*, page 4–12, New York, NY, USA. Association for Computing Machinery.
- Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, volume 30, pages 4765–4774. Curran Associates, Inc.
- Luo, K., Sanner, S., Wu, G., Li, H., and Yang, H. (2020a). Latent linear critiquing for conversational recommender systems. In *Proceedings of The Web Conference 2020, WWW '20*, pages 2535–2541, New York, NY, USA. Association for Computing Machinery.

## Bibliography

---

- Luo, K., Yang, H., Wu, G., and Sanner, S. (2020b). Deep critiquing for vae-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 1269–1278, New York, NY, USA. Association for Computing Machinery.
- Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.
- Lyu, S., Rana, A., Sanner, S., and Bouadjenek, M. R. (2021). A workflow analysis of context-driven conversational recommendation. In *Proceedings of the Web Conference 2021*, WWW '21, page 866–877, New York, NY, USA. Association for Computing Machinery.
- MacKenzie, I., Meyer, C., and Noble, S. (2013). How retailers can keep up with consumers. *McKinsey & Company*, 18:1.
- Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26*.
- Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- Martins, A. F. T. and Astudillo, R. F. (2016). From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1614–1623. JMLR.org.
- McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 165–172, New York, NY, USA. ACM, ACM.
- McAuley, J., Leskovec, J., and Jurafsky, D. (2012). Learning attitudes and attributes from multi-aspect reviews. In *2012 IEEE 12th International Conference on Data Mining, ICDM '12*, pages 1020–1025, Washington, DC, USA.
- McAuley, J., Targett, C., Shi, Q., and van den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 43–52, New York, NY, USA. Association for Computing Machinery.
- McCarthy, K., Salem, Y., and Smyth, B. (2010). Experience-based critiquing : reusing critiquing experiences to improve conversational recommendation. *Bichindaritz, I. and Montani, S. (eds.). Case-based reasoning research and development : 18th International Conference on Case-based Reasoning, ICCBR 2010, Alessandria, Italy, July 2010 : proceedings*.

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.
- Milenkoski, M., Antognini, D., and Musat, C. (2021). Recommending burgers based on pizza preferences: Addressing data sparsity with a product of experts. In *Proceedings of the First Workshop of Cross-Market Recommendation at RecSys 2021 (XMRec 2021)*.
- Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.
- Milojkovic, N., Antognini, D., Bergamin, G., Faltings, B., and Musat, C. (2020). Multi-gradient descent for multi-objective recommender systems. In *Proceedings of the AAAI (2020) - Workshop on Interactive and Conversational Recommendation Systems (WICRS)*.
- Mitreviski, B., Filipovic, M., Antognini, D., Lejal Glaude, E., Faltings, B., and Musat, C. (2021). Momentum-based gradient methods in multi-objective recommendation. In *Proceedings of the First Workshop on Multi-Objective Recommender Systems at RecSys 2021 (MORS 2021)*.
- Mnih, A. and Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.
- Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.
- Mukherjee, S. and Awadallah, A. (2020). Uncertainty-aware self-training for few-shot text classification. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21199–21212. Curran Associates, Inc.
- Musat, C.-C. and Faltings, B. (2015). Personalizing product rankings using collaborative filtering on opinion-derived topic profiles. In *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15*, page 830–836. AAAI Press.
- Musat, C.-C., Liang, Y., and Faltings, B. (2013). Recommendation using textual opinions. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI '13*, page 2684–2690. AAAI Press.
- Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.
- Ni, J., Li, J., and McAuley, J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

## Bibliography

---

- Ni, J. and McAuley, J. (2018). Personalized review generation by expanding phrases and attending on aspect-aware representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 706–711, Melbourne, Australia. Association for Computational Linguistics.
- Nikolakopoulos, A. N., Berberidis, D., Karypis, G., and Giannakis, G. B. (2019). Personalized diffusions for top-n recommendation. In *the 13th ACM Conference on Recommender Systems*, pages 260–268.
- Niu, Y., Jiao, F., Zhou, M., Yao, T., Xu, J., and Huang, M. (2020). A self-training method for machine reading comprehension with soft evidence extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3916–3927, Online. Association for Computational Linguistics.
- Padh, K., Antognini, D., Lejal Glaude, E., Faltings, B., and Musat, C. (2021). Addressing fairness in classification with a model-agnostic multi-objective algorithm. In Peters, J. and Sontag, D., editors, *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence (UAI)*, Proceedings of Machine Learning Research. PMLR.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Pappas, N. and Popescu-Belis, A. (2014). Explaining the stars: Weighted multiple-instance learning for aspect-based sentiment analysis. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 455–466, Doha, Qatar. Association for Computational Linguistics.
- Paranjape, B., Joshi, M., Thickstun, J., Hajishirzi, H., and Zettlemoyer, L. (2020). An information bottleneck approach for controlling conciseness in rationale extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1938–1952, Online. Association for Computational Linguistics.
- Pariser, E. (2011). *The filter bubble: What the Internet is hiding from you*. Penguin UK.
- Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2227–2237, New Orleans.

- Petrescu, D. A., Antognini, D., and Faltings, B. (2021). Multi-step critiquing user interface for recommender systems. In *Fifteenth ACM Conference on Recommender Systems*, RecSys '21, page 760–763, New York, NY, USA. Association for Computing Machinery.
- Pruthi, D., Gupta, M., Dhingra, B., Neubig, G., and Lipton, Z. C. (2020). Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online. Association for Computational Linguistics.
- Pu, P. and Chen, L. (2005). Integrating tradeoff support in product search tools for e-commerce sites. In *Proceedings of the 6th ACM conference on Electronic commerce*.
- Pu, P. and Chen, L. (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.
- Pu, P. and Faltings, B. (2000). Enriching buyers' experiences: The smartclient approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 289–296, New York, NY, USA. Association for Computing Machinery.
- Pu, P., Viappiani, P., and Faltings, B. (2006). Increasing user decision accuracy using suggestions. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 121–130.
- Quint, E., Wirka, G., Williams, J., Scott, S., and Vinodchandran, N. (2018). Interpretable classification via supervised variational autoencoders and differentiable decision trees.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*.
- Reddi, S., Kale, S., and Kumar, S. (2018). On the convergence of adam and beyond. In *International Conference on Learning Representations (ICLR)*.
- Reilly, J., McCarthy, K., McGinty, L., and Smyth, B. (2005). Explaining compound critiques. *Artificial Intelligence Review*.
- Reilly, J., Zhang, J., McGinty, L., Pu, P., and Smyth, B. (2007). Evaluating compound critiquing recommenders: A real-user study. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, EC '07, pages 114–123, New York, NY, USA. Association for Computing Machinery.
- Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, USA. AUAI Press.

## Bibliography

---

- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1278–II–1286. JMLR.org.
- Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '16, pages 1135–1144, New York, NY, USA. Association for Computing Machinery.
- Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer.
- Rohrbach, A., Hendricks, L. A., Burns, K., Darrell, T., and Saenko, K. (2018). Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.
- Rosasco, L., Vito, E. D., Caponnetto, A., Piana, M., and Verri, A. (2004). Are loss functions all the same? *Neural Computation*, 16(5):1063–1076.
- Sachdeva, N. and McAuley, J. (2020). How useful are reviews for recommendation? a critical review and potential improvements. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 1845–1848, New York, NY, USA. Association for Computing Machinery.
- Sedhain, S., Bui, H., Kawale, J., Vlassis, N., Kveton, B., Menon, A. K., Bui, T., and Sanner, S. (2016). Practical linear models for large-scale one-class collaborative filtering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 3854–3860.
- Sedhain, S., Menon, A. K., Sanner, S., and Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 111–112, New York, NY, USA. Association for Computing Machinery.
- Serrano, S. and Smith, N. A. (2019). Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy.
- Shi, Y., N, S., Paige, B., and Torr, P. (2019). Variational mixture-of-experts autoencoders for multi-modal deep generative models. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., and Diakopoulos, N. (2016). Grand challenges for hci researchers. *Interactions*, 23(5):24–25.



- Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153, International Convention Centre, Sydney, Australia. PMLR.
- Sinha, R. and Swearingen, K. (2002). The role of transparency in recommender systems. In *CHI'02 extended abstracts on Human factors in computing systems*, CHI EA '02, pages 830–831, New York, NY, USA. Association for Computing Machinery.
- Smith-Renner, A., Fan, R., Birchfield, M., Wu, T., Boyd-Graber, J., Weld, D. S., and Findlater, L. (2020). No explainability without accountability: An empirical study of explanations and feedback in interactive ml. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA. Association for Computing Machinery.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- Sun, P., Wu, L., Zhang, K., Fu, Y., Hong, R., and Wang, M. (2020). Dual learning for explainable recommendation: Towards unifying user preference prediction and review generation. In *Proceedings of The Web Conference 2020*, WWW '20, page 837–847, New York, NY, USA. Association for Computing Machinery.
- Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia. PMLR.
- Sutter, T., Daunhawer, I., and Vogt, J. (2020). Multimodal generative learning utilizing jensen-shannon-divergence. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6100–6110. Curran Associates, Inc.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *the IEEE conference on computer vision and pattern recognition*.
- Tenenbaum, J. B. (1999). *A Bayesian framework for concept learning*. PhD thesis, Massachusetts Institute of Technology.
- Tintarev, N. and Masthoff, J. (2015). *Explaining Recommendations: Design and Evaluation*, pages 353–382. Springer.

## Bibliography

---

- Torrens, M., Faltings, B., and Pu, P. (2002). Smart clients: Constraint satisfaction as a paradigm for scaleable intelligent information systems. *Constraints*. Kluwer Academic Publishers, Hingham, MA, USA., 7(1):49–69.
- Tsai, Y.-H. H., Liang, P. P., Zadeh, A., Morency, L.-P., and Salakhutdinov, R. (2019). Learning factorized multimodal representations. In *International Conference on Learning Representations (ICLR)*.
- van der Lee, C., Gatt, A., van Miltenburg, E., Wubben, S., and Krahmer, E. (2019). Best practices for the human evaluation of automatically generated text. In *International Conference on Learning Representations*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Viappiani, P., Faltings, B., and Pu, P. (2006). Preference-based search using example-critiquing with suggestions. *Journal of Artificial Intelligence Research (JAIR)*, pages 465–503.
- Viappiani, P., Pu, P., and Faltings, B. (2008). Preference-based search with adaptive recommendations. *Ai Communications*, 21(2-3):155–175.
- Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.
- Vyas, D. A., Eisenstein, L. G., and Jones, D. S. (2021). Hidden in plain sight—reconsidering the use of race correction in clinical algorithms. *Obstetrical & Gynecological Survey*, 76(1):5–7.
- Wall, E., Ghorashi, S., and Ramos, G. (2019). Using expert patterns in assisted interactive machine learning: A study in machine teaching. In *Interact 2019*.
- Wang, K., Hua, H., and Wan, X. (2019). Controllable unsupervised text attribute transfer via editing entangled latent representation. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Wang, S. and Manning, C. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 90–94, Jeju Island, Korea.
- Wiegrefe, S. and Marasovic, A. (2021). Teach me to explain: A review of datasets for explainable natural language processing. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

- Wilkinson, D., Alkan, O., Liao, Q. V., Mattetti, M., Vejsbjerg, I., Knijnenburg, B. P., and Daly, E. (2021). Why or why not? the effect of justification styles on chatbot recommendations. volume 39, New York, NY, USA. Association for Computing Machinery.
- Williams, M. D. and Tou, F. N. (1982). Rabbit: an interface for database access. In *Proceedings of the ACM Conference*, pages 83–87.
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.
- Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280.
- Wu, G., Luo, K., Sanner, S., and Soh, H. (2019a). Deep language-based critiquing for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, page 137–145, New York, NY, USA. Association for Computing Machinery.
- Wu, G., Volkovs, M., Soon, C. L., Sanner, S., and Rai, H. (2019b). Noise contrastive estimation for one-class collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 135–144, New York, NY, USA. Association for Computing Machinery.
- Wu, M. and Goodman, N. (2018). Multimodal generative models for scalable weakly-supervised learning. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Wu, Y., DuBois, C., Zheng, A. X., and Ester, M. (2016). Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 153–162, New York, NY, USA. Association for Computing Machinery.
- Xu, B., Wang, N., Chen, T., and Li, M. (2015a). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015b). Show, attend and tell: Neural image caption generation with visual attention. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France. PMLR.
- Yala, A., Lehman, C., Schuster, T., Portnoi, T., and Barzilay, R. (2019). A deep learning mammography-based model for improved breast cancer risk prediction. *Radiology*, 292(1):60–66.
- Yang, H., Shen, T., and Sanner, S. (2021). *Bayesian Critiquing with Keyphrase Activation Vectors for VAE-Based Recommender Systems*, page 2111–2115. Association for Computing Machinery, New York, NY, USA.

## Bibliography

---

- Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.
- Yao, L., Peng, N., Weischedel, R., Knight, K., Zhao, D., and Yan, R. (2019). Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7378–7385.
- Yelp (2016). Yelp open dataset 2016 - <https://www.yelp.com/dataset>.
- Yeomans, M., Shah, A., Mullainathan, S., and Kleinberg, J. (2019). Making sense of recommendations. *Journal of Behavioral Decision Making*, 32(4):403–414.
- Yin, Y., Song, Y., and Zhang, M. (2017). Document-level multi-aspect sentiment classification as machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2044–2054, Copenhagen, Denmark. Association for Computational Linguistics.
- Yu, M., Chang, S., Zhang, Y., and Jaakkola, T. (2019). Rethinking cooperative rationalization: Introspective extraction and complement control. 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)*, pages 4094–4103, Hong Kong, China. Association for Computational Linguistics.
- Yu, M., Zhang, Y., Chang, S., and Jaakkola, T. (2021). Understanding interlocking dynamics of cooperative rationalization. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc.
- Zaheer, M., Ahmed, A., and Smola, A. J. (2017). Latent lstm allocation: Joint clustering and non-linear dynamic modeling of sequence data. In *International Conference on Machine Learning*, pages 3967–3976.
- Zhang, J. and Curley, S. P. (2018). Exploring explanation effects on consumers' trust in online recommender agents. *International Journal of Human–Computer Interaction*, 34(5):421–432.
- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.
- Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., and Ma, S. (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*, SIGIR '14, page 83–92, New York, NY, USA. Association for Computing Machinery.

- Zhang, Y., Marshall, I., and Wallace, B. C. (2016). Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 795–804, Austin, Texas. Association for Computational Linguistics.
- Zhao, Q., Chen, J., Chen, M., Jain, S., Beutel, A., Belletti, F., and Chi, E. H. (2018). Categorical-attributes-based item classification for recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18*, page 320–328, New York, NY, USA. Association for Computing Machinery.
- Zhou, B., Sun, Y., Bau, D., and Torralba, A. (2018). Interpretable basis decomposition for visual explanation. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 122–138, Cham. Springer International Publishing.



# Diego Antognini

RESEARCH SCIENTIST IN MACHINE LEARNING & NATURAL LANGUAGE PROCESSING

✉ diegoantognini@gmail.com | 💻 www.diegoantognini.com

🏠 Diego Antognini | 🗣️ Diego999 2.5k\* | 🌐 diegoantognini | 🇨🇭 Swiss citizen



## Summary

6+ years of research experience in natural language processing (NLP), machine learning (ML), and multi-objective recommendation systems. Experienced in developing efficient models for on-device inference (order of one MB model size and one ms latency) and interpretable models that generate personalized and actionable textual explanations. Supervised 60+ B/M.Sc. projects/theses. Assessed 50+ student projects. Offering consulting services in NLP and ML.

## Experience

### Research Scientist

May 2022 - present

- My research is centered around efficient machine learning for NLP: alternatives to transformer-based language models, novel cost-effective pre-training methods via self-supervision, or better leverage data with active learning. I build models in the order of one-megabyte size, targeted for edge use cases (smart watches or assistants) to provide on-device inference with similar performance and higher throughput than large models.
- Deployed models in the order of 1MB size and 1ms latency used in *IBM Deep Search* that analyzes in real time scientific publications and patents.
- Produced 3 papers in AI & ML leading venues and 2 patents, all under review. Served in the program committee for EMNLP '22 and RecSys '22.

IBM Research AI

Zürich, Switzerland

### Lecturer and Supervisor for M.Sc. theses in NLP

Apr. 2022 - present

- Teaching deep learning for natural language processing classes in the course "Computational Language Technologies" with Dr. Kornelia Papp.
- Supervised 3 M.Sc. theses in the field of NLP and associated with companies in the area of insurances and data visualization.

HSLU - Lucerne University of Applied Sciences

Luzern, Switzerland

### Consultant and Expert for B.Sc. & M.Eng. theses

Jun. 2015 - present

- Gave talks on multiple technologies and offered consulting services for applied research in industrial projects in Prof. Hatem Ghorbel's group.
- Assessed 30+ B.Sc./M.Eng. theses and was an expert for the AI and GPU/parallel programming exams (Prof. Hatem Ghorbel, Prof. Cédric Bilat).

HES-SO - HE-ARC - University of Applied Sciences

Neuchâtel, Switzerland

### Visiting Researcher in Prof. Julian McAuley's group

Jul. 2021 - Oct. 2021

- Created an unsupervised critiquing method and a generative language model to help users rewrite cooking recipes to satisfy dietary restrictions.

UCSD - University of California San Diego

San Diego, CA, U.S.A.

### Research/Teaching Assistant

May 2017 - Mar. 2022

- Helped in teaching the courses: intelligent agents (M.Sc.), introduction to natural language processing (M.Sc., 2x), artificial intelligence (B.Sc., 3x).
- Supervised 30+ B./M.Sc. semester projects & theses. Worked with the data analytics & AI research team in Swisscom (led by Dr. Claudiu Musat).

EPFL - École Polytechnique Fédérale de Lausanne

Lausanne, Switzerland

### Machine Learning/Data Mining M.Sc. thesis Intern

Sep. 2016 - Mars. 2017

- Deployed a system to extract concepts from large corpora and build interactive knowledge graphs for invention developers to gain new insights.

Iprova GmbH

EPFL Innovation Park, Lausanne, Switzerland

## Education

### Ph.D. in Computer Science

May 2017 - Mar. 2022

- Advisor:** Prof. Boi Faltings. **Group:** the Artificial Intelligence Lab (LIA). **Program Committee & Reviewer:** \*ACL 20-22; EMNLP 20-21; RecSys '21.
- Award:** Won \$9,750 in the *IARPA Geopolitical Forecasting Challenge 2018*. Press coverage: *EPFL News (English)* and *24 Heures (French)*.
- Thesis** 📄: Textual Explanations and Critiques in Recommendation Systems. It solves two challenges: explanation generation and making them actionable (critiquing). It focuses on text understanding, explainable AI, and conversational recommendation. Fastest to graduate in the AI lab.
- Wrote 16 papers in AI & ML leading venues. 14 published: 7 conference, 6 workshop, and 1 demo. 2 papers currently under review at EMNLP 2022.
- Implemented the 1<sup>st</sup> PyTorch graph attention network and its sparse variant. Starred and forked on Github 2.7k+, and still 8k+ views per month.

EPFL - École Polytechnique Fédérale de Lausanne

Lausanne, Switzerland

### M.Sc. in Computer Science

Sep. 2014 - Apr. 2017

- Focus on NLP, AI, ML, and distributed systems. It includes a mandatory extra year of 62 ECTS credits to be accepted in the program. **GPA:** 5.5/6.0.
- Thesis:** From Relation Extraction to Knowledge Graphs (6.0/6.0). Developed a model that extracts concepts from large corpora and classifies the semantic relationship between them. It outperformed state-of-the-art models by 0.9 F1-score in the relation-classification task of SemEval-2010.

EPFL - École Polytechnique Fédérale de Lausanne

Lausanne, Switzerland

### B.Sc. in Computer Science

Sep. 2011 - Aug. 2014

- Major software engineering. **Awards:** 1) Excellent thesis and 2<sup>nd</sup> best GPA; 2) Derserving student in 2<sup>nd</sup> year. **GPA** 5.6/6.0.
- Thesis:** Computing Brain Neuronal Maps. Developed a multi-GPUs algorithm to compute an accurate 3D real-time rendering of the brain's electromagnetic activities. A speedup of ~100,000x has been achieved, reducing the computation time from 20h to 700ms.

HES-SO - HE-ARC - University of Applied Sciences

Neuchâtel, Switzerland

## Publications (selected)

### Interacting with Explanations through Critiquing 📄

Diego Antognini, Claudiu Musat, Boi Faltings

15 ICAI 2021

### Fast Critiquing with Self-Supervision for VAE-based Recommender Systems 📄

Diego Antognini and Boi Faltings

RecSys 2021

## Multi-Dimensional Explanation of Target Variables from Documents

Diego Antognini, Claudiu Musat, Boi Faltings

AAAI 2021

## Rationalization through Concepts

Diego Antognini and Boi Faltings

ACL 2021

Findings

## Addressing Fairness in Classification with a Model-Agnostic Multi-Objective Algorithm

Kirtan Padh, Diego Antognini, Emma L. Glaude, Boi Faltings, Claudiu Musat

UAI 2021

## Multi-Gradient Descent for Multi-Objective Recommender Systems

Nikola Milojkovic, Diego Antognini, Giancarlo Bergamin, Boi Faltings, Claudiu Musat

AAAI 2020

Workshop

## HotelRec: a Novel Very Large-Scale Hotel Recommendation Dataset

Diego Antognini and Boi Faltings

LREC 2020

## Talks

### Textual Explanations and Critiques in Recommendation Systems

2022

EPFL, Lausanne.

### Deep Learning with PyTorch

HES-SO - HE-ARC – University of Applied Sciences, Neuchâtel.

### Transformers for Time-Series predictions

2021

HES-SO - HE-ARC – University of Applied Sciences, Neuchâtel.

### Fast Critiquing with Self-Supervision for VAE-based Recommender Systems

RecSys 2021, Amsterdam.

### Rationalization through Concepts

ACL 2021, Online.

### Interacting with Explanations through Critiquing

1) IJCAI 2021, Online; 2) University of Toronto, Online; 3) Swisscom Lab, Online.

### Multi-Dimensional Explanation of Target Variables from Documents

AAAI 2021, Online.

### T-RECS: a Recommender Generating Explanations and Integrating Critiquing

2020

ECAI 2020, Online.

### Multi-Dimensional Explanation of Ratings from Reviews

1) University of Zürich, Zürich; 2) NLP Meetup, Zürich; 3) Swisscom AI, Online.

### 1) What are attention mechanisms?; 2) Overview of principles & models in recommender systems

HES-SO - HE-ARC – University of Applied Sciences, Neuchâtel.

### Learning to Create Sentence Semantic Relation Graphs for Multi-Document Summarization

2019

EMNLP 2019, Hong-Kong.

### Introduction to 1) PyTorch, 2) Spark, 3) [Contextualized] Word Embeddings, 4) Dialogue Systems

HES-SO - HE-ARC – University of Applied Sciences, Neuchâtel.

### Constraint Satisfaction Problem

2018

Artificial intelligence course at EPFL, Lausanne.

### Introduction to ML & Insights on Alpha Go (Zero)

High school du Bugnon, Lausanne.

### From Relation Extraction to Knowledge Graphs

2017

1) NLP Meetup, Zürich; 2) NLP course at EPFL, Lausanne; 3) HE-Arc, Neuchâtel.

## Skills

**Software** Efficient: Python, PyTorch, spaCy, Tensorflow, Spark, Bash. Prior Experience: C++, CUDA, Java, SQL, PHP, HTML/CSS.

**Miscellaneous** UNIX/Linux, Git, PyCharm, IntelliJ IDEA, Astah, Balsamiq Mockups, Visual Studio, Eclipse.

## Interests

**Languages** French (native), English (fluent), German (conversational), Spanish (conversational), Italian (conversational).

**Sport** I ride motorbikes in spring or surf on the coast during summer. In-between, I go four times per week to the gym to stay fit.

**Interests** I regularly read self-development books and question myself to become a better person. I have traveled to 25 countries through 4 continents; I would like to keep going and discover new cultures. Finally, I love dancing salsa and bachata.

## References

### Dr. Cristiano Malossi

Principal Research Manager

IBM Research AI, Switzerland

Current manager

✉ ACM@zurich.ibm.com

### Dr. Claudiu Musat

Research Manager

Google, Switzerland

Close collaborator

✉ claudiu.musat@gmail.com

### Prof. Boi Faltings

Full CS Professor

EPFL, Switzerland

Thesis advisor

✉ boi.faltings@epfl.ch

### Prof. Julian McAuley

Full CS Professor

University of California, San Diego

Research stay host

✉ jmcauley@ucsd.edu