HotelRec: a Novel Very Large-Scale Hotel Recommendation Dataset

Diego Antognini, Boi Faltings
Artificial Intelligence Laboratory
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
firstname.lastname@epfl.ch

Abstract

Today, recommender systems are an inevitable part of everyone's daily digital routine and are present on most internet platforms. State-of-the-art deep learning-based models require a large number of data to achieve their best performance. Many datasets fulfilling this criterion have been proposed for multiple domains, such as Amazon products, restaurants, or beers. However, works and datasets in the hotel domain are limited: the largest hotel review dataset is below the million samples. Additionally, the hotel domain suffers from a higher data sparsity than traditional recommendation datasets and therefore, traditional collaborative-filtering approaches cannot be applied to such data. In this paper, we propose HotelRec, a very large-scale hotel recommendation dataset, based on TripAdvisor, containing 50 million reviews. To the best of our knowledge, HotelRec is the largest publicly available dataset in the hotel domain ($50M$ versus $0.9M$) and additionally, the largest recommendation dataset in a *single domain* and with *textual reviews* ($50M$ versus $22M$). We release HotelRec for further research: https://github.com/Diego999/HotelRec.

Keywords: reviews, recommender systems, text mining, sentiment analysis

1. Introduction

The increasing flood of information on the web creates a need for selecting content according to the end user's preferences. Today, recommender systems are deployed on most internet platforms and play an important role in everybody's daily digital routine, including e-commerce websites, social networks, music streaming, or hotel booking.

Recommender systems have been investigated over more than thirty years (Bobadilla et al., 2013). Over the years, many models and datasets in different domains and various sizes have been developed: movies (Harper and Konstan, 2016), Amazon products (McAuley et al., 2015; He and McAuley, 2016), or music (Celma, 2010). With the tremendous success of large deep learning-based recommender systems, in better capturing user-item interactions, the recommendation quality has been significantly improved (Covington et al., 2016).

However, the increase in recommendation performance with deep learning-based models comes at the cost of large datasets. Most recent state-of-the-art models, such as (Wang et al., 2019), (Liang et al., 2018), or (He et al., 2017) necessitate large datasets (i.e., millions) to achieve high performance.

In the hotel domain, only a few works have studied hotel recommendation, such as (Wang et al., 2011) or (Zhang et al., 2015). Additionally, to the best of our knowledge, the largest publicly available hotel review dataset contains $870k$ samples (Li et al., 2016). Unlike commonly used recommendation datasets, the hotel domain suffers from higher data sparsity and therefore, traditional collaborative-filtering approaches cannot be applied (Zhang et al., 2015; Khaleghi et al., 2018; Musat and Faltings, 2015). Furthermore, rating a hotel is different than traditional products, because the whole experience lasts longer, and there are more facets to review (Khaleghi et al., 2018).

In contrast, we propose in this work HotelRec, a novel large-scale hotel recommendation dataset based on hotel reviews from TripAdvisor, and containing approximately 50 million reviews. A sample review is shown in Figure 1. To the best of our knowledge, HotelRec is the largest publicly available hotel review dataset (at least 60 times larger than previous datasets). Furthermore, we analyze various aspects of the HotelRec dataset and benchmark the performance of different models on two tasks: rating prediction and recommendation performance. Although reasonable performance is achieved by a state-of-the-art method, there

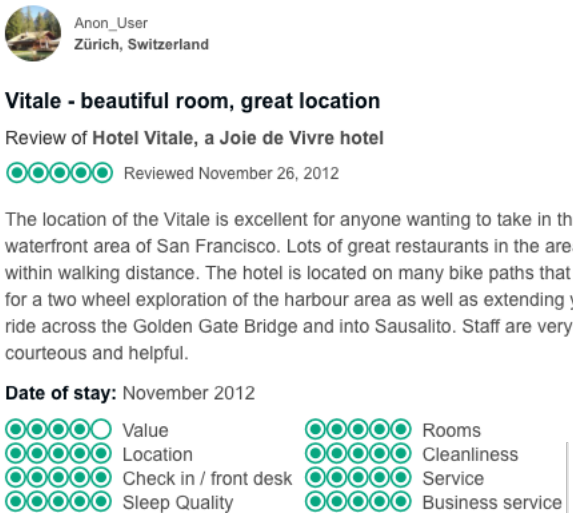


Figure 1: Review from TripAdvisor, with sub-ratings.

HotelRec	#Users	#Items	#Interactions	Sparsity
Full	21 891 294	365 056	50 264 531	99.99937%
5-core	2 012 162	312 081	21 108 245	99.99664%
20-core	72 603	38 903	2 222 373	99.92132%

Table 1: Statistics of the whole HotelRec dataset and its k-core subsets (number of users, items, interactions, and the sparsity ratio).

is still room for improvement. We believe that HotelRec will offer opportunities to apply and develop new large recommender systems, and push furthermore the recommendation for hotels, which differs from traditional datasets.

2. Related Work

Recommendation is an old problem that has been studied from a wide range of areas, such as Amazon products (McAuley and Leskovec, 2013), beers (McAuley et al., 2012), restaurants[1], images (Geng et al., 2015), music (Celma, 2010), and movies (Harper and Konstan, 2016). The size of the datasets generally varies from hundreds of thousands to tens of millions of user-item interactions; an interaction always contains a rating and could have additional attributes, such as a user-written text, sub-ratings, the date, or whether the review was helpful. At the time of writing, and to the best of our knowledge, the largest available recommendation corpus on a specific domain and with textual reviews, is based on Amazon Books and proposed by He and McAuley (2016). It contains a total of 22 million book reviews. In comparison, HotelRec has 2.3 times more reviews and is based on hotels. Consequently, HotelRec is the largest domain-specific public recommendation dataset *with textual reviews* and *on a single domain*. We highlight *with textual reviews*, because some other datasets (e.g., Netflix Prize (Bennett and Lanning, 2007)) contain more interactions, that *only* includes the rating and the date.

To the best of our knowledge, only a few number of datasets for hotel reviews have been created: 35k (Wang et al., 2011), 68k (Musat et al., 2013), 140k (Antognini et al., 2019), 142k (Cozza et al., 2018), 235k (Wang et al., 2011), 435k (Musat and Faltings, 2015), and 870k (Li et al., 2016). However, the number of users, items, and interactions is limited compared to traditional recommendation datasets. In contrast, the HotelRec dataset has at least two orders of magnitude more examples. Statistics of HotelRec is available in Table 1.

3. HotelRec

Everyday a large number of people write hotel reviews on on-line platforms (e.g., Booking[2], TripAdvisor[3]) to share their opinions toward multiple aspects, such as their *Overall* experience, the *Service*, or the *Location*. Among the most popular platforms, we selected TripAdvisor: according to their third quarterly report of November 2019[4], on the *U.S. Securities and Exchange Commission* website[5], TripAdvisor is the world's largest online travel site with approximately 1.4 million hotels. Consequently, we created our dataset HotelRec based on TripAdvisor hotel reviews. The statistics of the HotelRec dataset, the 5-core, and 20-core versions are shown in Table 1; each contains at least k reviews for each user or item.

In this section, we first discuss about the data collection process (Section 3.1.), followed by general descriptive statistics (Section 3.2.). Finally, Section 3.3. analyzes the overall rating and sub-ratings.

3.1. Data Collection

We first crawled all areas listed on TripAdvisor's SiteIndex[6]. Each area link leads to another page containing different information, such as a list of accommodations, or restaurants; we gathered all links corresponding to hotels. Our robot then opened each of the hotel links and filtered out hotels without any review. In total, in July 2019, there were 365 056 out of 2 502 140 hotels with at least one review.

Although the pagination of reviews for each hotel is accessible via a URL, the automatic scraping is discouraged: loading a page takes approximately one second, some popups might appear randomly, and the robot will be eventually blocked because of its speed. We circumvented all these methods by mimicking a human behavior with the program *Selenium*[7], that we have linked with *Python*[8]. However, each action (i.e., disabling the calendar, going to the next page of reviews) had to be separated by a time gap of one second. Moreover, each hotel employed a review pagination system displaying only five reviews at the same time, which majorly slowed down the crawling.

An example review is shown in Figure 1. For each review, we collected: the URL of the user's profile and hotel, the date, the overall rating, the summary (i.e., the title of the review), the written text, and the multiple sub-ratings when provided. These sub-ratings correspond to a fine-grained evaluation of a specific aspect, such as *Service*, *Cleanliness*, or *Location*. The full list of fine-grained aspects is available in Figure 1, and their correlation in Section 3.3.

We naively parallelized the crawling on approximately 100 cores for two months. After removing duplicated reviews, as in McAuley and Leskovec (2013), we finally collected 50 264 531 hotel reviews.

3.2. Descriptive Statistics

HotelRec includes 50 264 531 hotel reviews from TripAdvisor in a period of nineteen years (from February 1, 2001 to May 14, 2019). The distribution of reviews over the years is available in Figure 2d. There is a significant activity increase of users from 2001 to 2010. After this period, the number of reviews per year grows slowly and oscillates between one to ten million.

In total, there are 21 891 294 users. The distribution of reviews per user is shown in Figure 2a. Similarly to other recommender datasets (He and McAuley, 2016; Meyffret et al., 2012), the distribution resembles a Power-law distribution: many users write one or a few reviews. In HotelRec, 67.55% users have written only one review, and 90.73% with less than five reviews. Additionally, in the 5-core subset, less than 15% of 2 012 162 users had a peer with whom

[1] https://www.yelp.com/dataset/challenge

[2] https://www.booking.com/

[3] https://www.tripadvisor.com/

[4] https://www.sec.gov/ix?doc=/Archives/edgar/data/1526520/000156459019041094/trip-10q_20190930.htm

[5] https://www.sec.gov

[6] https://www.tripadvisor.com/SiteIndex

[7] https://selenium.dev/

[8] Using the package https://github.com/SeleniumHQ/selenium/

they have co-rated three or more hotels. Finally, the average user has 2.24 reviews, and the median is 1.00.

Relating to the items, there are 365 056 hotels, which is roughly 60 times smaller than the number of users. This ratio is also consistent with other datasets (McAuley and Leskovec, 2013; McAuley et al., 2012).

Figure 2b displays the distribution of reviews per hotel. The distribution also has a shape of a Power-law distribution, but its center is closer to 3 000 than the 100 of the user distribution. However, in comparison, only 0.26% hotels have less than five reviews and thus, the average reviews per hotel and the median are higher: 137.69 and 41.00.

Finally, we analyze the distribution of words per review, to understand how much people write about hotels. The distribution of words per review is shown in Figure 2c. The average review length is 125.57 words, which is consistent with other studies (McAuley and Leskovec, 2013).

3.3. Overall and Sub-Ratings

When writing a review, the *Overall* rating is mandatory: it represents the evaluation of the whole user experience towards a hotel. It is consequently available for all reviews in HotelRec. However, sub-ratings only assess one or more particular aspects (up to eight), such as *Service*, *Cleanliness*, or *Location*. Additionally, they are optional: the user can choose how many and what aspects to evaluate. Among all the reviews, 35 836 414 (71.30%) have one or several sub-ratings, with a maximum of eight aspects. The distribution of the number of assessed fine-grained aspects is shown in Table 2, where *All* represents the coverage over the whole set of reviews, and *With Sub-Ratings* over the set of reviews having sub-ratings (i.e., approximately 35 million). Interestingly, most of the sub-ratings are evaluated in a group of three or six aspects. We hypothesize that this phenomenon came from a limitation of TripAdvisor on the user interface, where the set of aspects to evaluate was predefined.

# Rated Aspects	Coverage (%)	
	All	With Sub-Ratings
1	0.404	0.566
2	0.893	1.252
3	29.474	41.341
4	2.220	3.113
5	5.201	7.295
6	31.982	44.858
7	1.120	1.572
8	0.002	0.002

Table 2: Statistics of the number of rated fine-grained aspects in the HotelRec dataset. Coverage is the ratio of reviews having i sub-ratings over: *All* reviews, and *only* reviews *With Sub-Ratings* available.

We analyze in Table 3 the distribution of the reviews with fine-grained and *Overall* ratings. Unsurprisingly, the *Overall* rating is always available as it is mandatory. In terms of aspects, there is a group of six that are majorly predominant (following the observation in Table 2), and two that are rarely rated: *Check-In* and *Business Service*. Surprisingly, these two aspects are not sharing similar rating averages and percentiles than the others. We explain this difference due to the small number of reviews rating them (approximately 2%). Furthermore, most ratings across aspects are positive: the 25^{th} percentile is 4, with an average of 4.23 and a median of 5.

Aspect	Coverage (%)	Average	25^{th}	50^{th}	75^{th}
Overall	100.00	4.15 ± 1.12	4	5	5
Service	99.27	4.29 ± 1.09	4	5	5
Check-In	2.73	4.00 ± 1.20	3	4	5
Business Serv.	1.69	3.65 ± 1.25	3	4	5
Location	71.22	4.40 ± 0.88	4	5	5
Value	73.63	4.12 ± 1.13	4	5	5
Cleanliness	73.69	4.33 ± 1.03	4	5	5
Rooms	70.92	4.12 ± 1.10	4	4	5
Sleep Quality	63.04	4.21 ± 1.08	4	5	5

Table 3: Descriptive statistics of the ratings of the *Overall* and fine-grained aspect ratings (e.g., *Service*, *Rooms*). Coverage describes the ratio of reviews having a particular fine-grained rating. The other columns represent the average, and the 25^{th}, 50^{th} (median), 75^{th} percentiles of the individual ratings.

Finally, in Figure 3, we computed the Pearson correlation of ratings between all pairs of aspects, including fine-grained and *Overall* ones. Interesting, all aspect-pairs have a correlation between 0.46 and 0.83. We observe that *Service*, *Value*, and *Rooms* correlate the most with the *Overall* ratings. Unsurprisingly, the aspect pair *Service-Check In* and *Rooms-Cleanliness* have a correlation of 0.80, because people often evaluate them together in a similar fashion. Interestingly, *Location* is the aspect that correlates the least with the others, followed by *Business Service*, and *Check-In*.

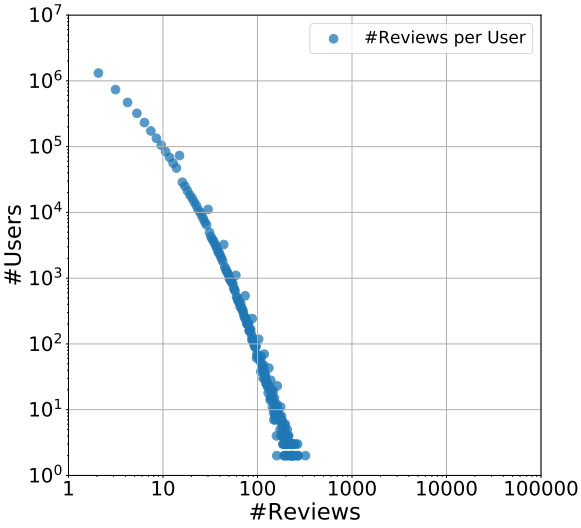
4. Experiments and Results

In this section, we first describe two different k-core subsets of the HotelRec dataset that we used to evaluate multiple baselines on two tasks: rating prediction and recommendation performance. We then detail the models we employed, and discuss their results.

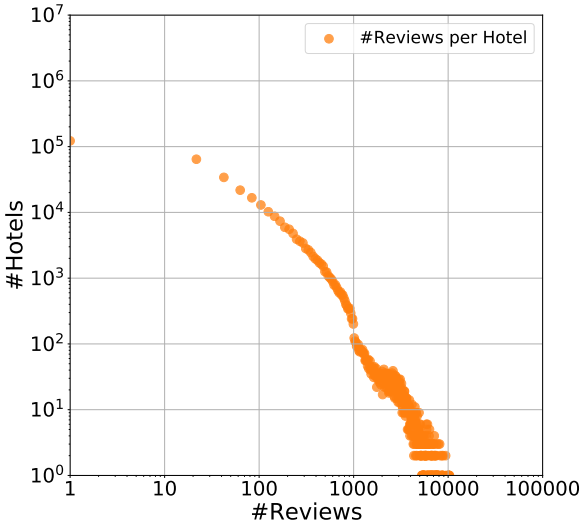
4.1. Datasets

We used the aforementioned dataset HotelRec, containing approximately 50 million hotel reviews. The characteristics of this dataset are described in Section 3.2. and Section 3.3. Following the literature (He et al., 2017; Cheng et al., 2018), we focused our evaluation on two k-core subsets of HotelRec, with at least k reviews for each user or item. In this paper, we employed the most common values for k: 5 and 20. We randomly divided each of the datasets into $80/10/10$ for training, validation, and testing subsets.

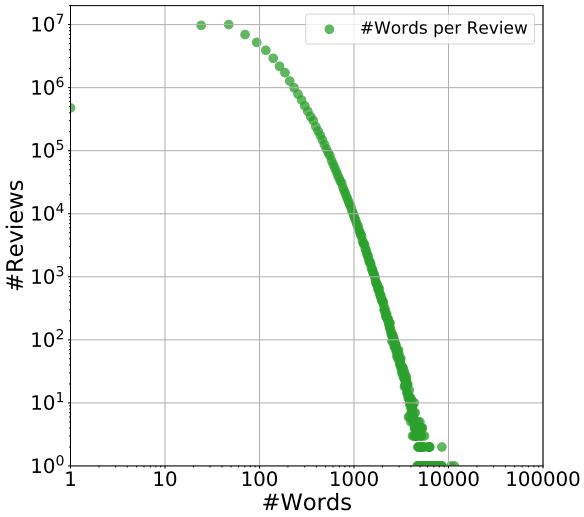
From each review, we kept the corresponding "userID", "itemID", rating (from 1 to 5 stars), written text, and date. We preprocessed the text by lowering and tokenizing it. Statistics of both subsets are shown in Table 1.



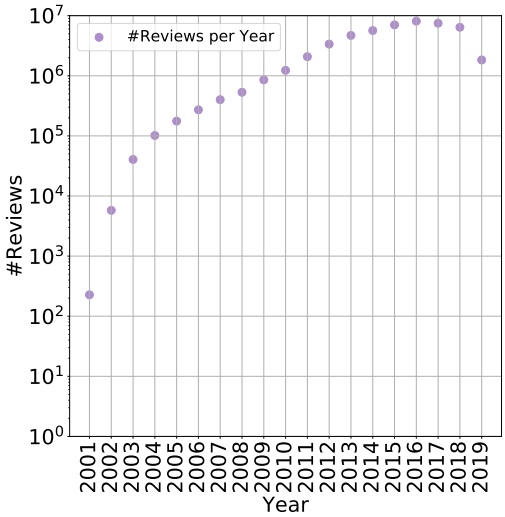
(a) Number of reviews per user. Mean: 2.24 ± 3.95.



(b) Number of reviews per item. Mean: 137.69 ± 342.94.



(c) Number of words per review. Mean: 125.57 ± 128.91.



(d) Number of reviews per year.

Figure 2: Histograms of multiple attributes of HotelRec, in logarithmic scales: number of reviews per user, item and year, and number of words per review.

4.2. Evaluation Metrics and Baselines

We evaluated different models on the HotelRec subsets, 5-core and 20-core, on two tasks: rating prediction and recommendation performance. We have separated the evaluation because most models are only tailored for one of the tasks but not both. Therefore, we applied different models for each task and evaluated them separately.

For the rating prediction task, following the literature, we reported the results in terms of Mean Square Error (MSE) and Root Mean Square Error (RMSE). We assessed the recommendation performance of a ranked list by Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) (He et al., 2015), as in He et al. (2017). We truncated the ranked list at 5, 10 and 20. The HR measures whether a new item is on the top-k list and NDCG measures the position of the hit by assigning higher scores to hits at top ranks. As in He et al. (2017), we computed both metrics for each test user and reported the average score. Regarding the models, we employed the following baselines:

- **Mean**: A simple model that predicts a rating by the mean ratings of the desired item. It is a good baseline in recommendation (Musat and Faltings, 2015);
- **HFT** (McAuley and Leskovec, 2013): A latent-factor approach combined with a topic model that aims to find topics in the review text that correlate with latent factors of the users and the items;
- **TransNet(-Ext)**: The model is based on Zheng et al. (2017), which learns a user and item profile based on former reviews using convolutional neural networks, and predicts the ratings using matrix factorization methods afterward. They added a regularizer network to improve performance. TransNet-Ext is an extension of TransNet by using a collaborative-filtering component in addition to user and item reviews history.

For the recommendation performance task, we used the following models :

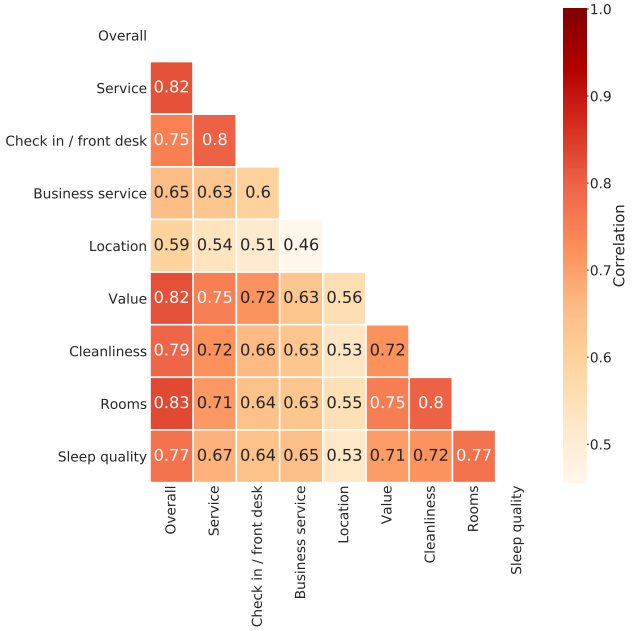


Figure 3: Pearson correlation between all fine-grained and overall ratings. All aspect pairs are highly correlated.

- **RAND**: A simple model recommending random items;
- **POP** (Rendle et al., 2009): Another non-personalized recommender method, where items are recommended based on their popularity (i.e., the number of interactions with users). It is a common baseline to benchmark the recommendation performance;
- **ItemKNN/UserKNN** (Sarwar et al., 2001): Two standard item-based (respectively user-based) collaborative filtering methods, using k nearest neighbors;
- **PureSVD** (Cremonesi et al., 2010): A similarity based approach that constructs a similarity matrix through the SVD decomposition of the rating matrix;
- **GMF** (He et al., 2017): A generalization of the matrix factorization method that applies a linear kernel to model the latent feature interactions;
- **MLP** (He et al., 2017): Similar than GMF, but it models the interaction of latent features with a neural network instead of a linear kernel;
- **NeuMF** (He et al., 2017): A model combining GMF and MLP to better model the complex user-item interactions.

Due to the large size of the HotelRec dataset, especially in the 5-core setting (around 20 million reviews), running an extensive hyper-parameter tuning for each *neural* model would require a high time and resource budget. Therefore, for the neural model, we used the default parameters from the original implementation and a random search of three trials. For all other models (i.e., HFT, ItemKNN, UserKNN, PureSVD), we ran a standard grid search over the parameter sets.

4.3. Rating Prediction

	Models	MSE	RMSE
5-core	Mean	0.7769	0.8814
	HFT - $K = 5$	0.7667	0.8756
	HFT - $K = 10$	**0.7520**	**0.8672**
	TransNet	0.8003	0.8946
	TransNet-Ext	0.8319	0.9121
20-core	Mean	**0.7546**	**0.8687**
	HFT - $K = 5$	0.7720	0.8786
	HFT - $K = 10$	0.7872	0.8872
	TransNet	0.8394	0.9162
	TransNet-Ext	1.0754	1.0370

Table 4: Evaluation of rating prediction in terms of Mean Square Error (MSE) and Root Mean Square Error (RMSE).

We show in Table 4 the performance in terms of the mean square error (MSE) and the root mean square error (RMSE). Surprisingly, we observe that the neural network TransNet and its extension perform poorly in comparison to the matrix factorization model HFT and the simple Mean baselines. Although TransNet learns a user and item profile based on the most recent reviews, it cannot capture efficiently the interaction from these profiles. Moreover, the additional collaborative-filtering component in TransNet-Ext seems to worsen the performance, which is consistent with the results of Musat et al. (2013); in the hotel domain, the set users who have rated the same hotels is sparser than usual recommendation datasets.
Interestingly, the Mean model obtains the best performance on the 20-core subset, while HFT achieves the best performance on the 5-core subset. We hypothesize that HFT and TransNet(-Ext) models perform better on the 5-core than 20-core subset, because of the number of data. More specifically, HFT employs Latent Dirichlet Allocation (Blei et al., 2003) to approximate topic and word distributions. Thus, the probabilities are more accurate with a text corpus approximately ten times larger.

4.4. Recommendation Performance

The results of the baselines are available in Table 5. HR@k and NDCG@k correspond to the Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG), evaluated on the top-k computed ranked items for a particular test user, and then averaged over all test users.
First, we can see that NeuMF significantly outperforms all other baselines on both k-core subsets. The other methods GMF and MLP - both used within NeuMF - also show quite strong performance and comparable performance. However, NeuFM achieves higher results by fusing GMF and MNLP within the same model. Second, if we compare ItemKNN and UserKNN, we observe that on both subsets, the user collaborative filtering approach underperform compared to its item-based variant, that matches the founding in the rating prediction task of the previous section, and the work of Musat et al. (2013; Musat and Faltings (2015). Additionally, PureSVD achieves comparable results with UserKNN.
Finally, the two non-personalized baselines RAND and

	Models	HR@5	NDCG@5	HR@10	NDCG@10	HR@20	NDCG@20
5-core	RAND	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000
	POP	0.0018	0.0007	0.0034	0.0010	0.0060	0.0014
	ItemKNN	0.0162	0.0072	0.0238	0.0088	0.0340	0.0103
	UserKNN	0.0118	0.0053	0.0176	0.0065	0.0248	0.0077
	PureSVD	0.0089	0.0039	0.0141	0.0050	0.0221	0.0064
	GMF	0.3899	0.2761	0.5340	0.3237	0.7055	0.3666
	MLP	0.4320	0.3070	0.5734	0.3533	0.7251	0.3915
	NeuMF	**0.4981**	**0.3589**	**0.6481**	**0.4066**	**0.7836**	**0.4401**
20-core	RAND	0.0004	0.0000	0.0010	0.0002	0.0019	0.0002
	POP	0.0064	0.0016	0.0109	0.0021	0.0210	0.0030
	ItemKNN	0.0236	0.0061	0.0411	0.0084	0.0682	0.0110
	UserKNN	0.0208	0.0054	0.0360	0.0073	0.0587	0.0095
	PureSVD	0.0216	0.0055	0.0375	0.0075	0.0616	0.0099
	GMF	0.3705	0.2565	0.5219	0.3047	0.6913	0.3477
	MLP	0.3731	0.2564	0.5251	0.3050	0.6962	0.3496
	NeuMF	**0.4274**	**0.3000**	**0.5776**	**0.3483**	**0.7354**	**0.3884**

Table 5: Evaluation of Top-K recommendation performance in terms of Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG).

POP obtain unsurprisingly low results, indicating the necessity of modeling user's preferences to a personalized recommendation.

5. Conclusion

In this work, we introduce HotelRec, a novel large-scale dataset of hotel reviews based on TripAdvisor, and containing approximately 50 million reviews. Each review includes the user profile, the hotel URL, the overall rating, the summary, the user-written text, the date, and multiple subratings of aspects when provided. To the best of our knowledge, HotelRec is the largest publicly available dataset in the hotel domain ($50M$ versus $0.9M$) and additionally, the largest recommendation dataset in *a single domain* and with *textual reviews* ($50M$ versus $22M$).

We further analyze the HotelRec dataset and provide benchmark results for two tasks: rating prediction and recommendation performance. We apply multiple common baselines, from non-personalized methods to competitive models, and show that reasonable performance could be obtained, but still far from results achieved in other domains in the literature.

In future work, we could easily increase the dataset with other languages and use it for multilingual recommendation. We release HotelRec for further research: `https://github.com/Diego999/HotelRec`.

6. Bibliographical References

Antognini, D., Musat, C., and Faltings, B. (2019). Multi-dimensional explanation of reviews. *arXiv preprint arXiv:1909.11386*.

Bennett, J. and Lanning, S. (2007). The netflix prize. In *Proceedings of the KDD Cup Workshop 2007*, pages 3–6, New York, August. ACM.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.

Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-based systems*, 46:109–132.

Celma, O. (2010). *Music Recommendation and Discovery in the Long Tail*. Springer.

Cheng, Z., Ding, Y., Zhu, L., and Kankanhalli, M. (2018). Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the 2018 World Wide Web Conference*, pages 639–648. International World Wide Web Conferences Steering Committee.

Covington, P., Adams, J., and Sargin, E. (2016). Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198. ACM.

Cozza, V., Petrocchi, M., and Spognardi, A. (2018). Mining implicit data association from tripadvisor hotel reviews. In Nikolaus Augsten, editor, *Proceedings of the Workshops of the EDBT/ICDT 2018 Joint Conference, Vienna, Austria, March 26, 2018*, volume 2083 of *CEUR Workshop Proceedings*, pages 56–61.

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*, pages 39–46. ACM.

Geng, X., Zhang, H., Bian, J., and Chua, T.-S. (2015). Learning image and user features for recommendation in social networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4274–4282.

Harper, F. M. and Konstan, J. A. (2016). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):19.

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*, pages 507–517. International World Wide Web Conferences Steering Com-

mittee.

He, X., Chen, T., Kan, M.-Y., and Chen, X. (2015). Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1661–1670. ACM.

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*, pages 173–182. International World Wide Web Conferences Steering Committee.

Khaleghi, R., Cannon, K., and Srinivas, R. (2018). A comparative evaluation of recommender systems for hotel reviews. *SMU Data Science Review*, 1(4):1.

Li, J., Monroe, W., and Jurafsky, D. (2016). Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

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*, pages 689–698. International World Wide Web Conferences Steering Committee.

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*, pages 165–172. ACM.

McAuley, J., Leskovec, J., and Jurafsky, D. (2012). Learning attitudes and attributes from multi-aspect reviews. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM '12, pages 1020–1025, Washington, DC, USA. IEEE Computer Society.

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*, pages 43–52. ACM.

Meyffret, S., Guillot, E., Médini, L., and Laforest, F. (2012). RED: a Rich Epinions Dataset for Recommender Systems. Research report, LIRIS.

Musat, C. C. and Faltings, B. (2015). Personalizing product rankings using collaborative filtering on opinion-derived topic profiles. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Musat, C. C., Liang, Y., and Faltings, B. (2013). Recommendation using textual opinions. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

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*, pages 452–461. AUAI Press.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA. ACM.

Wang, H., Lu, Y., and Zhai, C. (2011). Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–626. ACM.

Wang, X., He, X., Wang, M., Feng, F., and Chua, T.-S. (2019). Neural graph collaborative filtering. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 165–174, New York, NY, USA. ACM.

Zhang, K., Wang, K., Wang, X., Jin, C., and Zhou, A. (2015). Hotel recommendation based on user preference analysis. In *2015 31st IEEE International Conference on Data Engineering Workshops*, pages 134–138. IEEE.

Zheng, L., Noroozi, V., and Yu, P. S. (2017). Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 425–434. ACM.