ReviewGraph: A Knowledge Graph Embedding Based Framework For Review Rating Prediction With Sentiment Features

Bert de Vink, Natalia Amat-Lefort, Lifeng Han, Lifeng Han | Leiden University, NL

Universiteit Leiden
The Netherlands

The 16th IEEE International Conference on Knowledge Graphs
(ICKG) 2025 November 13-14, 2025 Limassol, Cyprus

Content

Introduction

Why ReviewGraph?

1. User generated hotel review star ratings affect up to 50% of booking decisions

2. Being able to predict user review star ratings makes it easier to understand what causes them to be at 4/5 instead of 5/5

3. It influences other metrics such as:
    - Occupancy rates
    - Revenue per room (RevPAR)
    - More abstract metrics such as brand reputation

4. Traditionally used methods (TF-IDF, BoW, Word2Vec) for review rating prediction miss important granular context, and simply use (co-)occurence of words

5. Current LLM based-methods are stuck at 60-70% accuracy, and lack explainability

6. ReviewGraph: combination of **sentiment, graph embeddings,** and **word co-occurrence**, sentiment is on a granular between words level. (Review can say bathroom is both clean and broken, mixed sentiment)

Related Work

Kumar, Mukesh, et al. "Efficient Hotel Rating Prediction from Reviews Using Ensemble Learning Technique." Wireless Personal Communications 137.2 (2024): 1161-1187.

What has been done before?

Embedding Technique	LRC	LRCV	SGDC	SVC	DTC	RFC	KNN
TF-IDF	0.60	**0.61**	0.57	0.57	0.53	0.58	0.53
BOW	0.60	0.60	0.57	0.57	0.38	0.59	0.53
Word2Vec	0.54	0.54	0.53	0.53	0.44	0.53	0.52

1. Kumar et al. (2024) is our baseline for this study
   - Used the same user review rating **dataset** (HotelRec Tripadvisor)
   - Used traditional NLP representation techniques
   - Used 7 different classifier models to compare; no LLMs
   - **Ensemble** learning using majority voting per representation technique
   - Results: 61% accuracy for best individual metric, 57% accuracy for best ensemble (BoW)
2. Novel LLM studies were able to accurately predict review rating scores with 67% accuracy, BERT-based models generally have an accuracy score of 60%
3. Only metric given for both of these studies was accuracy score, no sentiment was used

$$Accuracy = \frac{Correct\ Predictions}{All\ Predictions}$$

Research Questions

RQ1: How can **knowledge graph embeddings** and **sentiment** prediction models improve the accuracy of customer review-based overall star rating predictions?

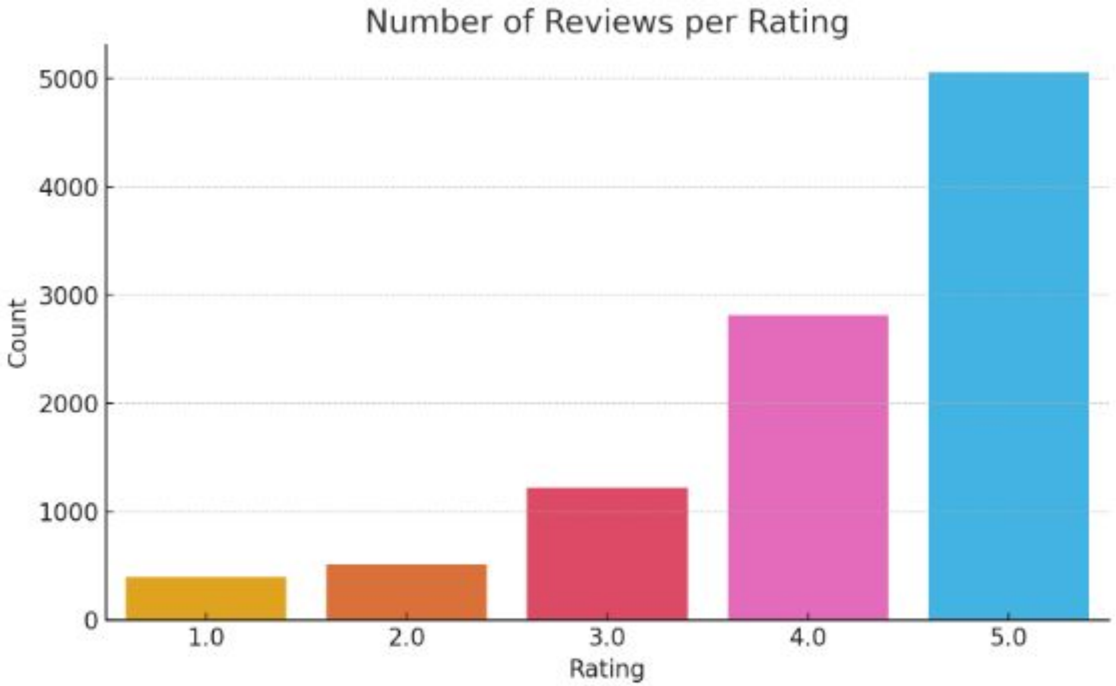
RQ2: How do state-of-the-art **LLMs** (GPT-4o) perform on review score prediction tasks, in comparison to task-specific NLP models?

Dataset

HotelRec Tripadvisor Dataset of 50 million reviews with review star ratings, used only the **first 10,000 reviews: data split**

- (train, test)=(80%, 20%) &
- 10-fold cross-validation (model robustness and de-bias)

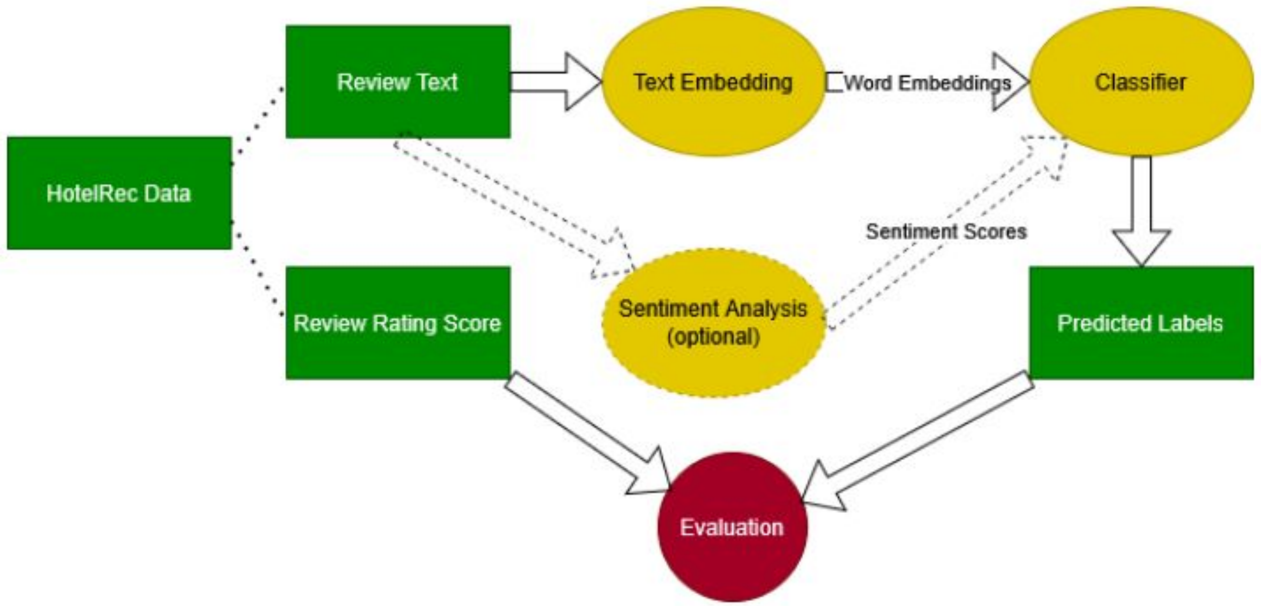
Class imbalance favors 5 star reviews -> we use oversampling to mitigate this



Methodology

Baseline

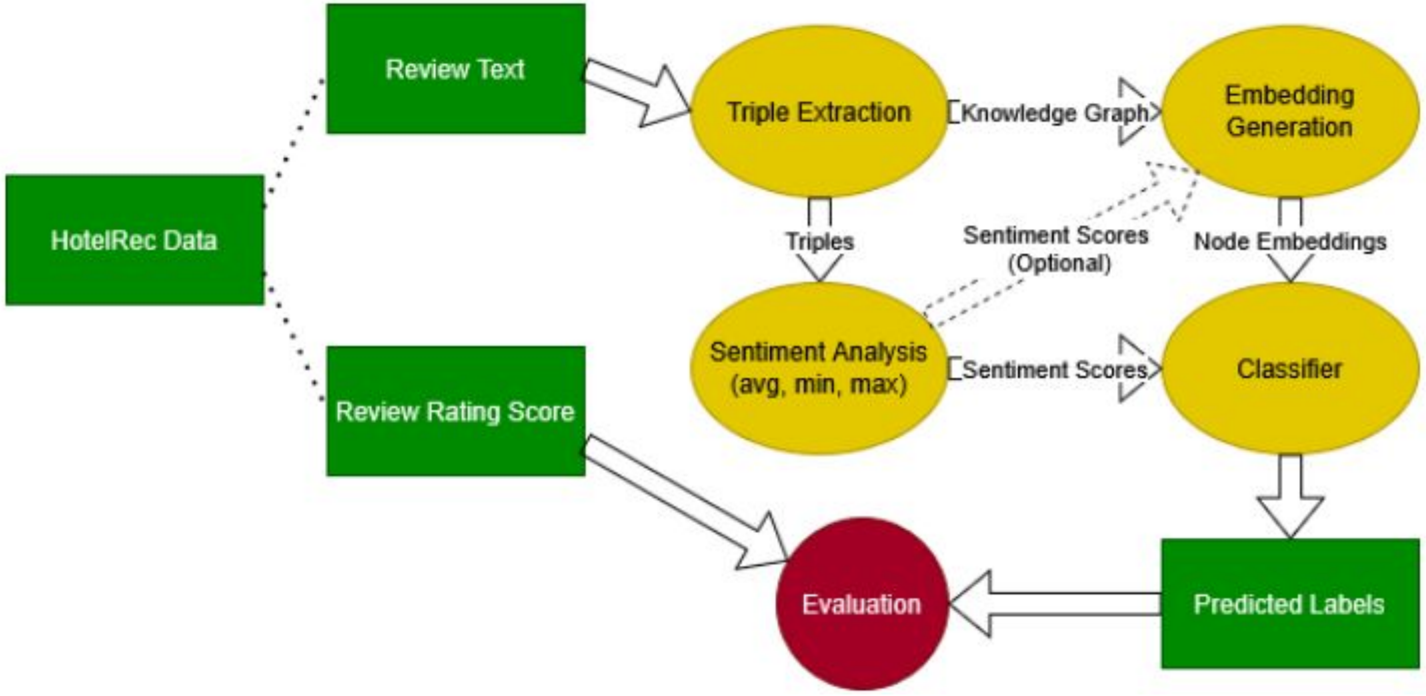
1. Not clear how much data Kumar et al (2024) used, we developed our own baseline to compare to instead
2. Means we can also use more metrics other than accuracy score, we chose: RMSE, MAE, and Cohen's Kappa
3. Used Word2Vec, TF-IDF, BoW
4. Random Forest, Logistic Regression Multi-Layer Perceptron, and Dummy (most common value) Classifier models



Methodology

ReviewGraph

1. **Triple** extraction using Stanford OpenIE, extracts subject, object, and relationship from sentences
2. Import those "triples" into knowledge graph, using Neo4J
3. Every "triple" has **sentiment** analysed, a value from -1 to 1 which is added to the relationship
4. Example:
   Bathroom –is (sentiment:0.6)-> Clean
5. Embeddings are generated using **Node2Vec**, represents graph as numbers for machine learning
6. Retrieve the <u>average, minimum, or maximum</u> sentiment of relationships connected to the review
7. Node2Vec and sentiment values are used to make prediction with classifier models



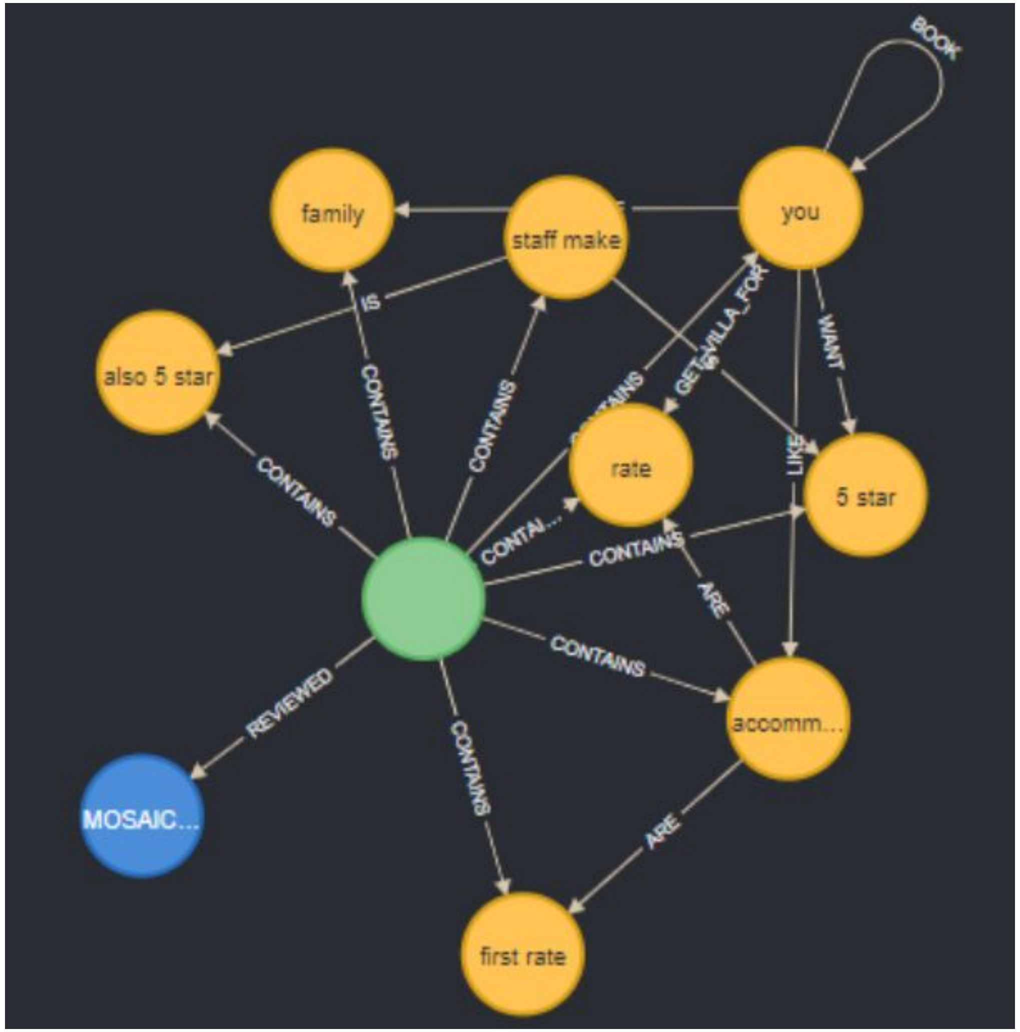
Methodology

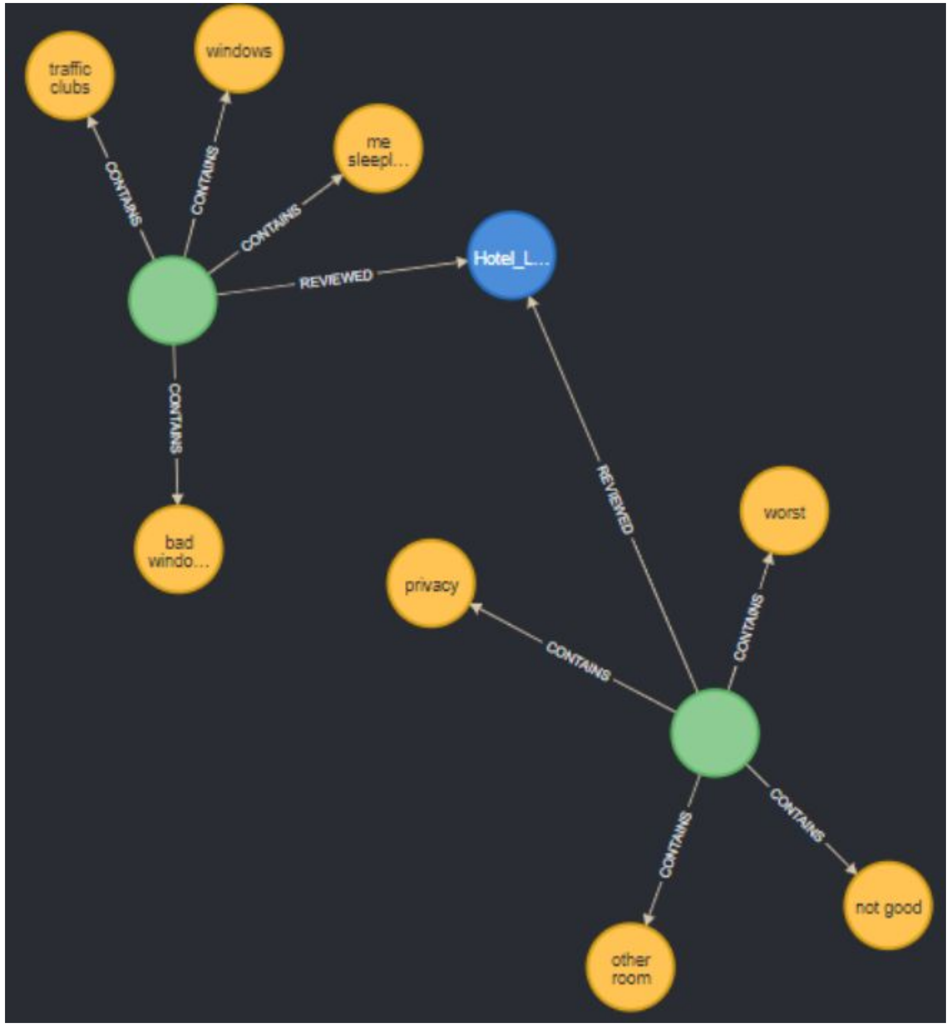
LLM

1. Simple prompt-based, GPT-4o was used
2. example-based learning: Given a training set of randomly selected 2,000 & 200 reviews from the same dataset
3. test on the rest of the whole data: Prompted to give predicted ratings - 9800 and 8000 testing.

"Our LLM-based model works by asking the LLM to predict the ratings of a set of reviews, based on the initial "training set" it is given. So we first give it 200, or 2000 reviews with their corresponding ratings, and then based on those it has to predict the ratings for the other 9800, or 8000 reviews we give it next. To accomplish this task the LLM is allowed to write its own code."

System Visualisation





extracted triples

Table 11: Sample of Extracted Triples from Review Data

Review ID	Subject	Relation	Object	Sentiment
144	Great pool	is in	wonderful spot by beach	0.83
3798	bed	was comfortable with	excellent linen	0.79
992	positive	were many small issues with	room for improvement	0.77
6254	loft	best feature of was	bathroom	0.64
276	Breakfast	was outstanding for	British fryup	0.61
1299	we	were	most impressed	0.53
2185	Our 40th school reunion weekend	was	help	0.40
1513	hotel	is well located in	historical center	0.27
4683	Hotel	was accommodating to	our group	0.00
5108	We	brought along	our 8yearold	0.00
721	We	had	5night stay	0.00
6958	Ravenna	was	crowded	0.00
3744	same	can	can said of bathroom	0.00
8153	This	is	our 4th year staying here	0.00
5882	manager	moved with	only minor change fee	0.00
9326	it	is	too much trouble	-0.40
6644	Poor excuse	is in	need	-0.42
5672	water temperature	keeps	fluctuating dangerously	-0.46
8623	check	is	wrong	-0.48
4844	My complaint	was	very poor wireless internet service	-0.68

Results 1

Baseline Results

1. Results with 10-Fold cross validation
2. Best model overall: TF-IDF with Logistic Regression, highest accuracy, lowest MAE
3. Best RMSE: BoW with Logistic Regression
4. Best Cohen's Kappa: Word2Vec with Logistic Regression

Classifier Model	Type	Accuracy	MAE	RMSE	Cohen's Kappa
Random Forest	Word2Vec	0.5942	0.5365	0.8971	0.3213
Logistic Regression	Word2Vec	**0.6286**	**0.4605**	**0.6981**	**0.4055**
MLP Classifier	Word2Vec	0.5461	0.5874	0.9336	0.2957
Dummy (Most Frequent)	Word2Vec	0.5060	0.8380	1.8686	0.0000
Random Forest	Bag of Words	0.5612	0.6751	1.3739	0.1888
Logistic Regression	Bag of Words	**0.6113**	**0.4608**	**0.6334**	**0.3855**
MLP Classifier	Bag of Words	0.6056	0.4746	0.6694	0.3751
Dummy (Most Frequent)	Bag of Words	0.5060	0.8380	1.8686	0.0000
Random Forest	TF-IDF	0.5619	0.6839	1.4245	0.1848
Logistic Regression	TF-IDF	**0.6367**	**0.4536**	0.6942	**0.4007**
MLP Classifier	TF-IDF	0.5966	0.4817	**0.6695**	0.3591
Dummy (Most Frequent)	TF-IDF	0.5060	0.8380	1.8686	0.0000

Results 2.1

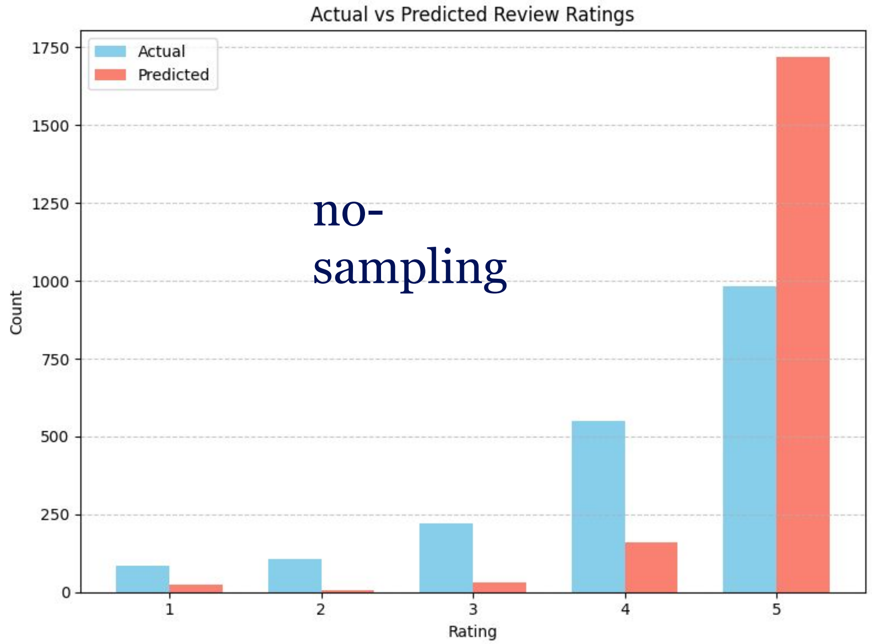
ReviewGraph Results

1. Results with 10-Fold cross validation
2. Performance substantially better in RMSE and Cohen's Kappa when using complex sentiment features (min/max)
3. Did other experiments, for Node2Vec embedding 5 dimensions performed best
4. Oversampling sacrifices other scores for Cohen's Kappa, because of class imbalance
5. Random Forest Classifier consistently performs best in most metrics for Word2Vec

Classifier Model	Type	Dim	Sampling	Accuracy	MAE	RMSE	Cohen's Kappa
Random Forest	Node2Vec	5	Oversampling	**0.4094**	**0.8766**	**1.6609**	0.0558
Logistic Regression	Node2Vec	5	Oversampling	0.3575	1.4730	4.3033	**0.0807**
Neural Network (MLP)	Node2Vec	5	Oversampling	0.2765	1.3625	3.2776	0.0601
Dummy (Most Frequent)	Node2Vec	5	Oversampling	0.0406	3.1612	11.1686	0.0000
Random Forest	Node2Vec	5	No Sampling	0.4771	**0.8121**	**1.6404**	**0.0400**
Logistic Regression	Node2Vec	5	No Sampling	**0.5085**	0.8355	1.8657	0.0073
Neural Network (MLP)	Node2Vec	5	No Sampling	0.4917	0.8190	1.7327	0.0254
Dummy (Most Frequent)	Node2Vec	5	No Sampling	0.5080	0.8388	1.8789	0.0000
Random Forest	N2V+Sentiment(min/max)	5	Oversampling	**0.5243**	**0.6379**	**1.0605**	0.2615
Logistic Regression	N2V+Sentiment(min/max)	5	Oversampling	0.4913	0.7569	1.4742	**0.2649**
Neural Network (MLP)	N2V+Sentiment(min/max)	5	Oversampling	0.4543	0.8389	1.6516	0.2249
Dummy (Most Frequent)	N2V+Sentiment(min/max)	5	Oversampling	0.0406	3.1612	11.1686	0.0000

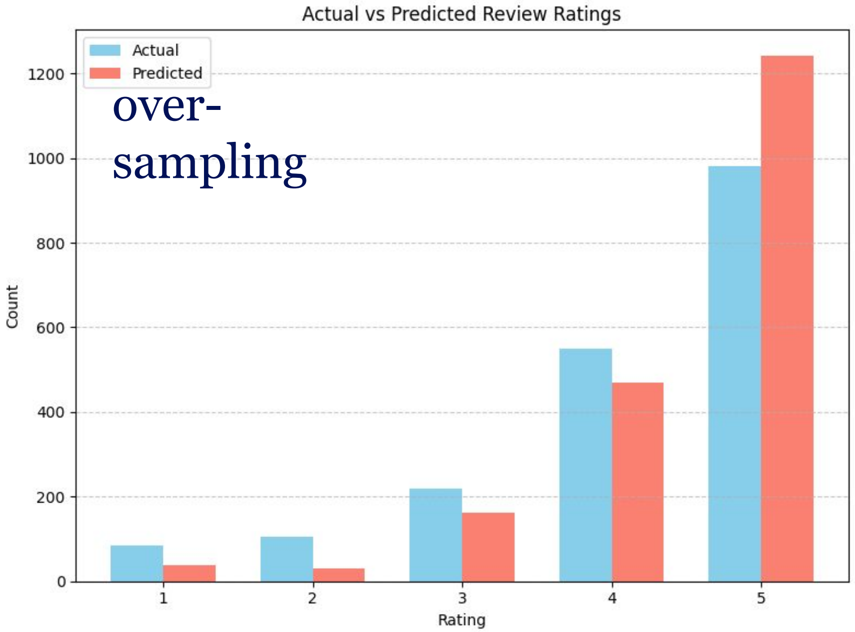
Results 2.2

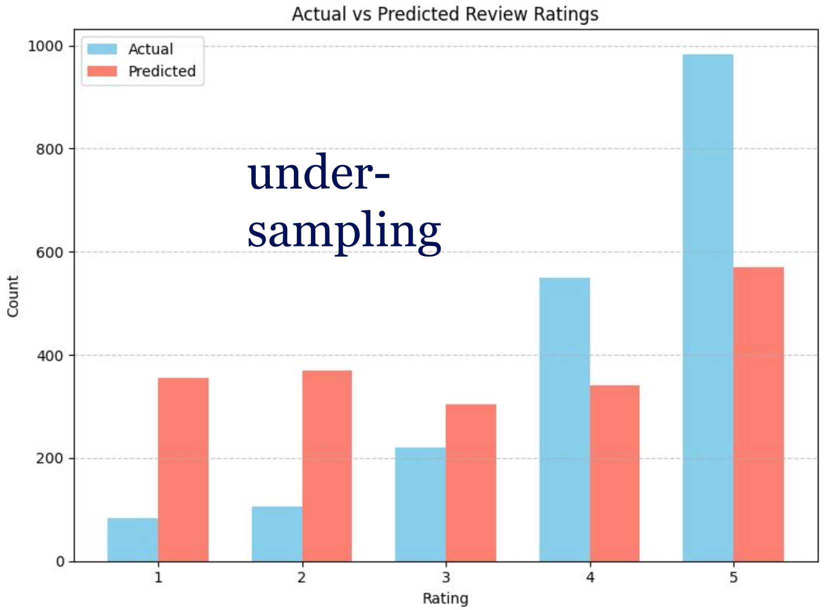
ReviewGraph Prediction Results



lower score: even lower
high score: even higher

lower score: not that lower
high score: not that higher





lower score: higher
higher score: lower
all: bring to the middle

Results 3

LLM Results

1. Results roughly on par with our baseline

Model	Sampling Size	Accuracy	MAE	RMSE	Cohen's Kappa
LLM Model	n=200	0.5165	0.8050	1.7748	0.0427
LLM Model	n=2000	**0.5867**	**0.5795**	**1.0622**	**0.2839**

1. example-based learning: Given a training set of randomly selected 200 & 2,000 reviews from the same dataset
2. test on the rest of the whole data: Prompted to give predicted ratings - 9800 and 8000 testing.

Results 4

Ablation Studies

1. We also tried doing the predictions with only the Node2Vec embeddings, and with only the sentiment features

2. We found that both perform worse individually than when combined together

3. Average sentiment drags down the accuracy of the model without min/max sentiment

Model	Sampling	Accuracy	MAE	RMSE	Cohen's Kappa
Random Forest	No Sampling	**0.4822**	**0.7692**	**1.4740**	**0.1374**
Logistic Regression	No Sampling	0.4256	1.1906	3.1855	0.1278
Neural Network (MLP)	No Sampling	0.3962	1.3818	4.0206	0.0906
Dummy (Most Frequent)	No Sampling	0.5059	0.8465	1.9201	0.0000
Random Forest	Oversampling	0.3921	**1.0170**	**2.2597**	0.1177
Logistic Regression	Oversampling	**0.4436**	1.1190	2.9150	**0.1403**
Neural Network (MLP)	Oversampling	0.3725	1.4292	4.1669	0.1112
Dummy (Most Frequent)	Oversampling	0.0433	3.1535	11.1484	0.0000

Table 9: Performance metrics of the ReviewGraph model without Node2Vec embeddings.

Model	Sampling	Accuracy	MAE	RMSE	Cohen's Kappa
Random Forest	No Sampling	**0.5425**	**0.6517**	**1.1741**	**0.2393**
Logistic Regression	No Sampling	0.3411	1.2494	3.1546	0.1482
Neural Network (MLP)	No Sampling	0.4575	1.1092	2.9588	0.0544
Dummy (Most Frequent)	No Sampling	0.5059	0.8465	1.9201	0.0000
Random Forest	Oversampling	**0.4992**	**0.7079**	**1.2705**	**0.2287**
Logistic Regression	Oversampling	0.3766	1.1659	2.9876	0.1642
Neural Network (MLP)	Oversampling	0.3009	1.0974	2.3400	0.0536
Dummy (Most Frequent)	Oversampling	0.0433	3.1535	11.1484	0.0000

Table 10: Performance metrics of ML models using only 5 dimensional Node2Vec embedding features, with and without oversampling.

All scores

Train test split (80-20) results

Classifier Model	Type	Dim	Sampling	Accuracy	MAE	RMSE	Cohen's Kappa
Random Forest	Node2Vec	100	Oversampling	0.4709	0.8135	1.6409	0.0352
Logistic Regression	Node2Vec	100	Oversampling	0.1968	1.7331	4.6121	0.0180
Neural Network (MLP)	Node2Vec	100	Oversampling	0.1757	1.3282	2.4874	-0.0025
Random Forest	Node2Vec	5	Oversampling	0.4060	0.8722	1.6388	0.0574
Logistic Regression	Node2Vec	5	Oversampling	0.2334	1.6249	4.3699	0.0307
Neural Network (MLP)	Node2Vec	5	Oversampling	0.1664	1.8537	5.0644	0.0043
Random Forest	Node2Vec	5	No Sampling	0.4673	0.8315	1.7105	0.0188
Logistic Regression	Node2Vec	5	No Sampling	0.3060	1.5281	4.2998	0.0562
Neural Network (MLP)	Node2Vec	5	No Sampling	0.2257	1.3735	3.2643	-0.0070
Dummy (Most Frequent)	Node2Vec	5	No Sampling	0.5059	0.8465	1.9201	0.0000
Random Forest	Node2Vec	100	No Sampling	0.4951	0.8542	1.9134	-0.0076
Logistic Regression	Node2Vec	100	No Sampling	0.3344	1.4776	4.2751	0.0751
Neural Network (MLP)	Node2Vec	100	No Sampling	0.2700	1.2081	2.6321	0.0049
Dummy (Most Frequent)	Node2Vec	100	No Sampling	0.5059	0.8465	1.9201	0.0000
Random Forest	N2V+Sentiment(min/max)	5	No Sampling	0.5590	0.5992	1.0185	0.2607
Logistic Regression	N2V+Sentiment(min/max)	5	No Sampling	0.3632	1.3060	3.5018	0.1688
Neural Network (MLP)	N2V+Sentiment(min/max)	5	No Sampling	0.4626	1.0752	2.8300	0.0937
Dummy (Most Frequent)	N2V+Sentiment(min/max)	5	No Sampling	0.5059	0.8465	1.9201	0.0000
Random Forest	N2V+Sentiment(min/max)	5	Oversampling	0.5404	0.6301	1.0896	0.2829
Logistic Regression	N2V+Sentiment(min/max)	5	Oversampling	0.4250	1.1345	3.0057	0.2107
Neural Network (MLP)	N2V+Sentiment(min/max)	5	Oversampling	0.5039	0.8238	1.7965	0.0862
Dummy (Most Frequent)	N2V+Sentiment(min/max)	5	Oversampling	0.0433	3.1535	11.1484	0.0000
Random Forest	Word2Vec	–	–	0.5705	0.5775	0.9805	0.2988
Logistic Regression	Word2Vec	–	–	**0.6035**	**0.5010**	**0.7920**	**0.3793**
Neural Network (MLP)	Word2Vec	–	–	0.5275	0.6325	1.0555	0.2828
Dummy (Most Frequent)	Word2Vec	–	–	0.4765	0.9025	2.0405	0.0000
Random Forest	Bag of Words	–	–	0.5350	0.7335	1.5405	0.1704
Logistic Regression	Bag of Words	–	–	0.5960	0.4855	0.6875	0.3802
Neural Network (MLP)	Bag of Words	–	–	0.5920	0.5010	0.7360	0.3698
Dummy (Most Frequent)	Bag of Words	–	–	0.4765	0.9025	2.0405	0.0000
Random Forest	TF-IDF	–	–	0.5265	0.7550	1.6010	0.1500
Logistic Regression	TF-IDF	–	–	0.5955	0.5120	0.7990	0.3474
Neural Network (MLP)	TF-IDF	–	–	0.5880	0.5015	0.7165	0.3607
Dummy (Most Frequent)	TF-IDF	–	–	0.4765	0.9025	2.0405	0.0000
LLM Model	LLM (n=200)	–	–	0.5165	0.8050	1.7748	0.0427
LLM Model	LLM (n=2000)	–	–	0.5867	0.5795	1.0622	0.2839

10-fold cross-validation eval:

Classifier Model	Type	Dim	Sampling	Accuracy	MAE	RMSE	Cohen's Kappa
Random Forest	Node2Vec	5	Oversampling	0.4094	0.8766	1.6609	0.0558
Logistic Regression	Node2Vec	5	Oversampling	0.3575	1.4730	4.3033	0.0807
Neural Network (MLP)	Node2Vec	5	Oversampling	0.2765	1.3625	3.2776	0.0601
Dummy (Most Frequent)	Node2Vec	5	Oversampling	0.0406	3.1612	11.1686	0.0000
Random Forest	Node2Vec	5	No Sampling	0.4771	0.8121	1.6404	0.0400
Logistic Regression	Node2Vec	5	No Sampling	0.5085	0.8355	1.8657	0.0073
Neural Network (MLP)	Node2Vec	5	No Sampling	0.4917	0.8190	1.7327	0.0254
Dummy (Most Frequent)	Node2Vec	5	No Sampling	0.5080	0.8388	1.8789	0.0000
Random Forest	N2V+Sentiment(min/max)	5	Oversampling	0.5243	0.6379	1.0605	0.2615
Logistic Regression	N2V+Sentiment(min/max)	5	Oversampling	0.4913	0.7569	1.4742	0.2649
Neural Network (MLP)	N2V+Sentiment(min/max)	5	Oversampling	0.4543	0.8389	1.6516	0.2249
Dummy (Most Frequent)	N2V+Sentiment(min/max)	5	Oversampling	0.0406	3.1612	11.1686	0.0000
Random Forest	Word2Vec	–	–	0.5942	0.5365	0.8971	0.3213
Logistic Regression	Word2Vec	–	–	0.6286	0.4605	0.6981	**0.4055**
MLP Classifier	Word2Vec	–	–	0.5461	0.5874	0.9336	0.2957
Dummy (Most Frequent)	Word2Vec	–	–	0.5060	0.8380	1.8686	0.0000
Random Forest	Bag of Words	–	–	0.5612	0.6751	1.3739	0.1888
Logistic Regression	Bag of Words	–	–	0.6113	0.4608	**0.6334**	0.3855
MLP Classifier	Bag of Words	–	–	0.6056	0.4746	0.6694	0.3751
Dummy (Most Frequent)	Bag of Words	–	–	0.5060	0.8380	1.8686	0.0000
Random Forest	TF-IDF	–	–	0.5619	0.6839	1.4245	0.1848
Logistic Regression	TF-IDF	–	–	**0.6367**	**0.4536**	0.6942	0.4007
MLP Classifier	TF-IDF	–	–	0.5966	0.4817	0.6695	0.3591
Dummy (Most Frequent)	TF-IDF	–	–	0.5060	0.8380	1.8686	0.0000

10-Fold Cross-Validation Performance comparison across all classifier models, feature types, sampli

Conclusion & Further Research

1. Current version of ReviewGraph performs worse than our own baseline
    - 63% accuracy and 0.4 Cohen's Kappa for Baseline vs 52% accuracy and 0.26 for our model
2. However, the results indicate significant promise. Room for improvement
3. Some possible improvements include:
    - Using inductive graph machine learning algorithms
    - Better triple extraction/knowledge graph extraction
    - Improved preprocessing and graph structure
4. Aside from these improvements, we also see potential in research the following:
    - GraphRAG methodology to quickly summarise 1000s of reviews
    - Removing/modifying nodes to make simulations of change average star rating scores
    - User studies and improving the visual utility of the graph structure

Questions & Answers

Ευχαριστώ

Universiteit
Leiden
The Netherlands

ReviewGraph: A Knowledge Graph Embedding Based Framework For Review Rating Prediction With Sentiment Features

https://github.com/aaronlifenghan/ReviewGraph (codes, visuralisation, interface software)

Universiteit Leiden
The Netherlands

Code & Supplementary materials

Appendix / GPT5-assisted summary

✅ **Problem & Motivation**

- In the hospitality industry, online customer reviews strongly influence business performance (booking decisions, occupancy, revenue, brand reputation). (arXiv)
- The challenge: predicting review ratings from textual reviews remains difficult, especially when using *interpretable*or *efficient* methods.

• • Existing approaches use Bag-of-Words, TF-IDF, Word2Vec, LLMs; many are **either** high-cost (LLMs) **or** low interpretability (basic NLP). The authors identify a gap for a method that is **both efficient and interpretable**. (arXiv)

🧩 Proposed Framework: ReviewGraph

- The paper proposes *ReviewGraph* (for Review Rating Prediction, RRP) which transforms the textual review into a **knowledge graph (KG)** representation:
    - Extract (subject, predicate, object) triples from the review text using an OpenIE method (Stanford CoreNLP OpenIE) rather than fine-tuned LLM triple extraction (for efficiency). (arXiv)
    - Associate **sentiment scores** with the extracted graph elements (e.g., for entities or relations) to capture the emotional valence of content.
- A graph embedding method (specifically **Node2Vec**) is used to generate node/edge embeddings from the constructed KG. Those embeddings become features in a downstream classifier for rating prediction. (arXiv)
- They also incorporate **sentiment features** (average, min, max sentiment scores) alongside the graph embedding features.

• • The resulting model integrates structure (KG embeddings) + sentiment + classification to predict review ratings.

📊 Baselines & Comparisons

- Baseline embedding strategies:
  - Bag of Words (BoW)
  - TF-IDF
  - Word2Vec
- They also compare to **LLM-based rating prediction** (they fine-tune or prompt an LLM on a subset of reviews) to see how their method stacks up.

• • The dataset: They use a subset of the “HotelRec” dataset (hotel reviews from TripAdvisor) — specifically ~10,000 reviews from ~59 hotels. (arXiv)

📈 Experimental Findings

- On metrics like Accuracy, MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), and **Cohen’s Kappa**(an agreement-based metric), ReviewGraph performed **similarly to the best LLM models**, but with **considerably lower computational cost** (no ensemble models needed). (arXiv)
- ReviewGraph outperformed the simpler baselines (BoW, TF-IDF, Word2Vec) especially on the agreement metric (Cohen’s Kappa), indicating stronger reliability of predictions.
- For example, the best performing ReviewGraph version used **5 embedding dimensions**, oversampling, and complex sentiment features (min, max, avg). Random Forest classifier worked best in many configurations. (arXiv)
- They argue the structured KG representation enables better interpretability and visual exploration (e.g., you can inspect the KG and see which entities/predicates drive low ratings) — which is less obvious with LLMs.
- They mention integration potential into Retrieval-Augmented Generation (RAG) systems; i.e., the KG output may be reused for further downstream generation or interactive analytics.

🔍 Key Contributions

1. **Conceptual**: Propose combining knowledge-graph representation of reviews (triples + sentiment) with graph embeddings for rating prediction.
2. **Methodological/Empirical**: Demonstrate on a hospitality-review dataset that this method can match LLM performance while improving interpretability and efficiency.
3. **Practical/Deployable**: Provide an open-source platform and output (GitHub link) so practitioners (e.g., hoteliers) can explore the KG outputs and integrate into dashboards or RAG workflows.

4. **Analytical insight**: They explore trade-offs between model complexity, interpretability, compute cost, and predictive performance in the review-rating context.

💡 Implications & Future Directions

- Graph-based embeddings (especially when enriched with sentiment) offer a promising middle ground: better than basic text embeddings, cheaper than full LLM ensembles, and more interpretable.
- For domains where interpretability matters (like business operations, hospitality management), KG-based models may be preferable to black-box deep networks.
- Future work: They suggest using **Graph Neural Networks (GNNs)** instead of Node2Vec, or **fine-tuned LLMs for triple extraction** (improved extraction quality) to boost performance further.
- Also, because the review KG potentially integrates with RAG (for interactive querying of review content), this opens avenues for **explainable analytics** not just prediction.
- Limitations: Only one dataset, relatively small subset; method relies on triple extraction (which may miss nuance); classification of rating may still smooth over fine distinctions.