Augmenting the User-Item Graph with Textual Similarity Models

Federico López*  
Heidelberg Institute for Theoretical Studies  
Heidelberg, Germany  
federico.lopez@h-its.org

Martin Scholz  
Google  
Mountain View, USA  
mscholz@google.com

Jessica Yung  
Google  
Zürich, Switzerland  
jessicayung@google.com

Marie Pellat  
Google  
Paris, France  
mpellat@google.com

Michael Strube  
Heidelberg Institute for Theoretical Studies  
Heidelberg, Germany  
michael.strube@h-its.org

Lucas Dixon  
Google  
Paris, France  
ldixon@google.com

ABSTRACT
This paper introduces a simple and effective form of data augmentation for recommender systems. A paraphrase similarity model is applied to widely available textual data – such as reviews and product descriptions – yielding new semantic relations that are added to the user-item graph. This increases the density of the graph without needing further labeled data. The data augmentation is evaluated on a variety of recommendation algorithms, using Euclidean, hyperbolic, and complex spaces, and over three categories of Amazon product reviews with differing characteristics. Results show that the data augmentation technique provides significant improvements to all types of models, with the most pronounced gains for knowledge graph-based recommenders, particularly in cold-start settings, leading to state-of-the-art performance.1

CCS CONCEPTS
• Information systems → Recommender systems.

KEYWORDS
recommender systems, data augmentation, knowledge graphs

1 INTRODUCTION
Recommender systems (RS) estimate users’ preferences for items to provide personalized recommendations and a better user experience. The underlying assumption is that users may be interested in items selected by people who share similar interactions with them. By mining interaction records, a model implicitly learns user-user and item-item similarities, and exploits them for recommendations [65].

One of the major challenges for RS is the long tail of users with sparse interaction data. Making recommendations for new users or about new items with little data is difficult (the so-called cold-start problem). To alleviate this issue, a considerable strand of research has explored incorporating side information to augment the interaction data [70]. A prominent branch of this work explores using textual descriptions and reviews, which often exist alongside rating or purchase data [52, 72]. However, a recent re-evaluation of such techniques indicates that the benefits are marginal, especially in cold-start scenarios. For example, modern deep learning-based models yield minimal changes in performance when reviews are masked [63].

In this paper we introduce a simple and effective form of data augmentation that leverages pre-trained textual semantic similarity models. The models are applied to widely available textual data, like product descriptions and reviews, yielding new relations between items. In this manner, we complement the implicit item similarity learnt from interactions by introducing explicit semantic relations based on textual attributes. We explore how these relations guide models to group semantically similar items and boost the recommendation system’s performance.

The data augmentation technique is evaluated on a variety of models where the user-item graph can naturally be extended with new relations. Many of these models are variants of knowledge graph recommenders since they provide an expressive and unified framework for modelling side information between users, items, and related entities [90]. Analysis of local and global geometric measures of the generated graphs indicate that the augmented graphs are better represented in hyperbolic spaces (see §4). Therefore, we also explore a variety of representational alternatives for recommender systems, including Euclidean [4, 84, 88], complex [69] and hyperbolic [3, 11] geometries.

Finally, we analyze how the proposed relations are more efficient at encoding semantic information present in textual descriptions compared to baselines that extract latent features from raw text. We find that our technique is more effective at reducing the generalization error, and this is particularly notable in cold-start settings. Furthermore, we analyze which type of text is more helpful for this task, noting that the product description can provide very condensed and useful information to draw semantic relations.

In this work we investigate two types of inductive biases: a data-dependent bias, by augmenting the graph relations via pre-trained language models, and a geometric bias, through the choice of a metric space to embed the relations. By means of a thorough assessment, we show how they complement each other. As we enlarge the data, the hyperbolic properties of the graph become more evident. In summary, we make the following contributions:

• We propose an unsupervised data augmentation technique by explicitly mining semantic relations derived from items’ text that boosts the performance, and is particularly effective in cold-start settings.

---

1Work partially done during an internship at Google.

2Code available at: https://github.com/PAIR-code/recommendation-rudders
• We provide a thorough analysis of local and global structural aspects of the user-item graph, and its augmented version, which indicates that the graph is better represented in hyperbolic space rather than in Euclidean.
• We explore KG methods developed in Euclidean, hyperbolic and complex spaces, and showcase how they achieve state-of-the-art performance for recommendations, when leveraged with the appropriate relations.

2 MINING SEMANTIC RELATIONS

2.1 Motivation
A popular approach to incorporate side information in recommender systems is to exploit user reviews and item descriptions, which often exist alongside rating or purchase data [52, 72]. A textual review is much more expressive than a single rating, and the underlying assumption is that reviews are effective user/item descriptors [23, 51], thus they can be used to learn better latent features [8, 13].

In a conventional deep learning architecture, these features are used in matrix factorization [94]. In such a setup, the model is burdened with the task of learning an implicit similarity function that should emerge from the textual input and the user’s purchase history. Based on this affinity, alike users and items are grouped and leveraged for recommendations. However, Sachdeva & McAuley [63] show that recent models yield minimal changes in performance when reviews are masked. They observe that reviews are more effective when used as a regularizer [31, 52], rather than as side data to extract latent features, and this behavior is accentuated in cold-start scenarios.

This paper introduces a different approach to benefit from textual attributes: it proposes leveraging advances in textual similarity models [9] to feed a recommender explicit content-based similarities via new edges in the interaction graph. This requires no supervision and complements the implicit similarity function that the model learns, without increasing the computational complexity. The data augmentation increases the density in the graph and serves as an efficient regularizer without the need to reduce the effective capacity of the model [30].

2.2 Method
The goal of the proposed method is to augment the user-item interaction graph with item-item relations. These relations are based on the semantic similarity between the item descriptions and reviews. Initially we collect all the available text for each item \( i \) in the set of items \( I \). This includes metadata, such as item name, descriptions and reviews. We experiment with various filters, as heuristic measures of the helpfulness of different types of text (e.g. top-k longest reviews, or only the metadata).

To compute text embeddings we employ the Universal Sentence Encoder (USE) [9], as it has shown good performance on sentence similarity benchmarks, and it can be applied without any further fine-tuning. Moreover, the average review length in purchase datasets tends to be one paragraph [1, 52], and USE has also been pre-trained to encode paragraphs composed of more than one sentence. We compute one embedding \( e_{ij} \in \mathbb{R}^d \) for each review (or descriptor) \( j \) corresponding to the item \( i \). The final embedding for item \( i \) is the result of taking the mean of all its review embeddings.

Once we have assigned an embedding to all items, we employ cosine similarity to compute the similarity between them. Finally, we extend the original user-item training set with the semantic relations between pairs of items. We filter out low-similarity pairs, and select the top-k highest similarities to be added as relations.

3 REPRESENTING THE GRAPH
Given that we extend the user-item graph by adding semantic relations, we need to account for the different types of edges present in the augmented graph. We model this multi-relational graph as a knowledge graph, since they offer a flexible approach to add multiple relations between diverse entities. In this section we describe knowledge graphs, and their application in recommender systems.

3.1 Knowledge Graphs
Knowledge graphs (KGs) are multi-relational graphs where nodes represent entities and typed-edges represent relationships among entities. They are popular data structures for representing heterogeneous knowledge in the shape of \((\text{head}, \text{relation}, \text{tail})\) triples, which can be queried and used in downstream applications. The usual approach to work with KGs is to learn representations of entities and relations as vectors, for some choice of space \( S \) (typically \( \mathbb{R} \)), such that the KG structure is preserved. More formally, let \( G = (E, R, T) \) be a knowledge graph where \( E \) is the set of entities, \( R \) is the set of relations and \( T \subset E \times R \times E \) is the set of triples stored in the graph. Most of the KG embedding methods learn vectors \( h, t \in \mathbb{R}^{\kappa E} \) for \( h, t \in E \), and \( r \in \mathbb{R}^{\kappa R} \) for \( r \in R \). The likelihood of a triple to be correct is evaluated using a model specific score function \( \phi : E \times R \times E \rightarrow \mathbb{R} \).

3.2 KG for Recommender Systems
Knowledge graph embedding methods\(^3\) have been widely adopted into the recommendation problem as an effective tool to model side information [27]. Multiple relations between users, items, and heterogeneous entities can be mapped into the KG and incorporated to alleviate data sparsity and enhance the recommendation performance [90].

Knowledge graph-based recommender systems can be seen as multi-task models, with well-established advantages. Learning several tasks (relations) at a time reduces the risk of over-fitting by generalizing the shared entity representations [92]. The data augmentation also improves the generalization of the model and acts as an effective regularizer [30]. Furthermore, akin relations can help the model to learn different types of entity interactions, such as similarities, that are finally exploited for recommendations [62].

Multi-relational knowledge graphs exhibit an intricate and varying structure as a result of the logical properties of the relationships they encode [43, 54, 68]. An item can be connected to different entities by symmetric, anti-symmetric, or hierarchical relations. To capture these non-trivial patterns more expressive operators become necessary.

\(^2\)An ablation of the heuristics and the encoder can be found in §6.4.

\(^3\)In this work we focus on embedding methods, according to the classification presented on [27], and not in path-based methods.
We augment the user-item graph by exploiting heterogeneous relations between items and other entities. Our starting point is the user-item graph, composed of solely (user, buy, item) triples. We augment this bipartite graph with the aforementioned semantic relations. These are (item_A, has_semantic_similarity, item_B) triples, meaning that there is a semantic overlap between the descriptors corresponding to these items. Previous work has explored relations between items and diverse entities such as product brands and categories [1, 86, 93], or movie directors and actors [7, 87, 90], depending on the dataset. To the best of our knowledge, this paper is the first to explore using pre-trained semantic similarity models to create new relations.

The graph augmentation modifies its size and structure. A change in its connectivity affects the optimal embedding space to operate [25]. Thus, we analyze this aspect in the next section.

### 4 Interaction Graph Analysis

The predominant approach to deal with graphs has been to embed them in an Euclidean space. Nonetheless, graphs from many diverse domains exhibit non-Euclidean features [5]. In particular, several RS datasets exhibit power-law degree distributions [10], and properties of scale-free networks [6, 36], which imply a latent hyperbolic geometry [38]. If we match the geometry of the target embedding space to the structure of the data, we can improve the representation fidelity [25, 49]. With the goal of understanding which type of Riemannian manifold would be more suitable as embedding space, we analyze different structural aspects and geometric properties of the graph.

Our analysis shows that when we augment the relationships in the graph, the added edges modify its connectivity and structure, making it more hyperbolic-like.

### 4.1 Data

To investigate the recommendation problem with regard to different relationships and geometries, we focus on the Amazon dataset⁶ [52, 56], as it is a standard benchmark for RS, and it provides item reviews and metadata in the form of textual descriptions. Nonetheless, our analysis generalizes to RS datasets that exhibit a latent hyperbolic geometry, studied in [6, 10, 36].

Specifically, we adopt the 5-core split for the branches “Musical Instruments” (Musins), “Video Games” (VGames) and “Arts, Crafts and Sewing” (ArtsCrafts), which form a diverse dataset in size and domain. Besides the semantic relations, we also add relationships available on the dataset that have already been explored in previous work [1, 86, 93]. These are:

- **also_bought**: users who bought item A also bought item B.
- **also_view**: users who bought the item A also viewed item B.
- **category**: the item belongs to one or more categories.
- **brand**: the item belongs to one brand.

The number of each type of relation added to the final augmented graph is reported in Table 2.

---

⁶https://nijiamno.github.io/amazon/index.html
work, we analyze models operating in the $n$-dimensional Poincaré ball: $\mathbb{H}^n = \{ x \in \mathbb{R}^n : ||x|| < 1 \}$.\(^7\)

For two points $x, y \in \mathbb{H}^n$ the distance in this space is defined as:

$$d_{\mathbb{H}}(x, y) = \cosh^{-1} \left( 1 + \frac{||x - y||^2}{(1 - ||x||^2)(1 - ||y||^2)} \right) \quad (2)$$

### 4.3 Curvature Analysis

Curvature is a geometric property that describes the local shape of an object. If we draw two parallel paths on a surface with positive curvature like a sphere, these two paths move closer to each other while for a negatively curved surface like a saddle, these two paths tend to be apart. There are multiple notions of curvature in Riemannian manifolds, with varying granularity. In the interest of space,\(^8\) we only recall a key notion: hyperbolic spaces have constant negative curvature, Euclidean spaces have zero curvature (flat) and spherical spaces are positively curved.

\(^7\)Although \([11]\) learns the curvature, we fix it to $c = 1$

\(^8\)For an in-depth treatment see \([42]\).  

### 4.2 Hyperbolic Geometry

Hyperbolic geometry is a non-Euclidean geometry with constant negative curvature. Hyperbolic space is naturally equipped for embedding symbolic data with hierarchical structures \([57, 64]\). Intuitively, that is because the amount of space grows exponentially as points move away from the origin. This mirrors the exponential growth of the number of nodes in trees with increasing distance from the root \([15]\). Thus, hyperbolic space can be seen as the continuous analogue to a discrete tree-like structure. Embedding norm represents depth in the hierarchy, and distance between embeddings the affinity or similarity of the respective items \([48]\). In this work, we apply the Ollivier-Ricci curvature to analyze the graphs \([58]\). Since this type of curvature characterizes the space locally, we plot the results in Figure 1.

We can observe that nodes and edges in the user-item graphs exhibit a very negative curvature (in red color). Negatively curved edges are highly related to graph connectivity, and removing them would result in a disconnected graph \([55]\). As we add more relationships, the augmented graph becomes much more connected, therefore these edges play a less important role. Nonetheless, the overall curvature, as shown by the vast majority of nodes and edges, remains negative. The correspondence with the negative curvature of hyperbolic space suggests that both, the user-item and the augmented graphs would profit from a representation in that geometry, rather than in an Euclidean one.

### 4.4 $\delta$-hyperbolicity

Also known as Gromov hyperbolicity \([24]\), $\delta$-hyperbolicity quantifies with a single number the hyperbolicity of a given metric space. The smaller the $\delta$ is, the more hyperbolic-like or negatively-curved the space is. This measure has also been adapted to graphs \([21]\).

We report the $\delta$-mean and $\delta$-max in Table 3. We can see that both measures decrease when we compare the user-item (U-I) graph with the augmented one (Augmen). The metric shows that, as we add more relationships to the initial user-item graph, it becomes more hyperbolic-like. This global metric complements the local curvature and also indicates that the graph fits into a hyperbolic space.

| MusIns | VGames | Arts&Crafts |
|--------|--------|-------------|
| # Users | 27,530 | 55,223 | 56,210 |
| # Items | 10,620 | 17,408 | 22,931 |
| buy | 231,392 | 497,577 | 494,485 |
| semantic | 102,497 | 168,877 | 218,120 |
| also_bought | 37,396 | 395,817 | 436,931 |
| also_view | 24,811 | 278,150 | 49,512 |
| category | 45,716 | 168,877 | 218,120 |
| brand | 10,525 | 17,269 | 22,670 |
| Total | 452,337 | 1,446,963 | 1,217,068 |

Table 2: Number of users, items and each type of relations added on different branches of the Amazon dataset.

| MusIns | VGames | Arts&Crafts |
|--------|--------|-------------|
| Nodes | 38,380 | 40,396 | 72,651 |
| Edges | 219,156 | 496,237 | 473,427 |
| Density (%) | 0.015 | 0.025 | 0.009 |
| Avg degree | 11.5 | 20.1 | 13.0 |
| $\delta$-mean | 0.270 | 0.086 | 0.285 |
| $\delta$-max | 2 | 1.5 | 2 |

Table 3: Statistics of graphs derived from different branches of the Amazon dataset, without relations (user-item, "U-I") and with all relations (augmented, "Augmen"). $\delta$-mean and $\delta$-max refer to the $\delta$-hyperbolicity (lower is more hyperbolic).
5 EXPERIMENTS

We experiment with different approaches to represent the multi-relational interaction graph for the task of generating recommendations. Our aim is to compare recent KG techniques, with a particular focus on the ones operating in hyperbolic spaces, with KG methods applied in previous work and SotA recommender systems.

Baselines: The recommender system baselines are:

- **BPR** [61]: Standard collaborative filtering baseline for RS based on matrix factorization (MF) with Bayesian personalized ranking.
- **NeuMF** [29]: Fuses MF with a multi-layer perceptron to learn latent user/item features and model interactions. We enhance this model by initializing the item features with pre-trained embeddings.
- **CML++** [31]: Metric learning baseline that encodes interactions in a joint space using Euclidean distance. Following [63], we learn extra user and item biases.
- **HyperML** [79]: Hyperbolic adaptation of CML.

Since reproducing all the works that adopt KGs is unfeasible (see Table 1), and in cases such as [1, 90, 93] the KG method is applied practically without modifications, we directly employ the KG models themselves. The selected methods are:

- **TransE** [4]: Translation-based KG method.
- **TransH** [84]: Hyperplane translation-based KG method.
- **DistMult** [88]: Based on multi-linear product, a generalization of the dot product.
- **RotatE** [69]: Performs rotations in the complex plane C.
- **RotRef** [11]: Based on rotations and reflections.
- **MuR** [3]: Based on multiplications and additions.

For RotRef and MuR we compare to the Euclidean and hyperbolic versions.

Data. To evaluate the models we utilize the branches of the Amazon dataset introduced in §4.1. Since it is very costly to rank all the available items, following [29, 79], we randomly select 100 samples which the user has not interacted with, and rank the ground truth amongst these samples. To generate evaluation splits, the penultimate and last item the user has interacted with are withheld as dev and test sets respectively.

Implementation details: Given differences in preprocessing strategies, we reproduce the baselines based on their public implementations. To ensure consistency in the results we conduct a hyper-parameter search for all methods on the validation set. All models are trained with the Adam optimizer [35], and operate with 64 latent dimensions. We train for 1000 epochs with early stopping on the dev set. To optimize parameters in hyperbolic models we apply tangent space optimization [12]. We choose the best learning rate for each method from from [0.0001, 0.0005, 0.001, 0.005] and batch-size from [256, 512, 1024]. In each case, we report the average of three runs. Preliminary experiments with multi-task losses (splitting the loss between KG and RS components as in [7, 81, 87]) did not show significant improvements therefore we disregard this approach. Models that incorporate item features (NeuMF and CML++) are fed with the text embeddings used to compute the semantic similarities. In this way, all models have access to information extracted from the same sources.

Setup: We evaluate the models in two setups: **User-item**, where we only employ the user-item interactions, and **Augmented**, where we utilize the graph with all the added relationships. Moreover, for the Augmented case we follow the standard data extension protocol by adding inverse relations to the train split [11, 40]. That is, for each triple \((h, r, t)\), we also add the inverse \((t, r^{-1}, h)\).

Evaluation protocols and Metrics: To evaluate the recommendation performance of KG methods we only look at the buy relation. For each user \(u\) we rank the items \(i_j\) according to the scoring function \(\phi(u, buy, i_j)\). We adopt normalized discounted cumulative gain (DCG) and hit ratio (HR), both at 10, as well-established ranking evaluation metrics for recommendations.

Research Questions: Through our experiments we aim to answer the following questions:

- **RQ1** How do KG methods perform compared to recently published RS?
- **RQ2** What is the impact of the data augmentation?
- **RQ3** How important are different relations to improving recommendations?
- **RQ4** Which text attributes are most helpful?
- **RQ5** How do different metric spaces compare?

6 RESULTS AND DISCUSSION

6.1 RQ1: Performance over user-item graph

In Table 4 we report the results for all models. Regarding the user-item results, we can observe that NeuMF is a very strong baseline, surpassing the performance of all other RS, and several KG methods. This can be explained by the fact that NeuMF and CML++ have access to more information than other baselines, since they are fed with embeddings generated from the items’ text. However, RotRef and MuR outperform all models. These models are designed to deal with multi-relational graphs, but in this case they only see one type of relation (buy). Although they operate solely based on the user-item interactions, the compound operators that they incorporate allow them for a more expressive representation of the bipartite graph, which results in improved recommendations. KG methods applied only on the user-item graph report a very high performance due to their enhanced representation capacity, thus we consider they should be adopted as hard-to-beat baselines in further research in recommender systems.

6.2 RQ2: Exploiting Augmented Data

We can see that all models (except for HyperML in MusIns) have a significant boost in performance when we train them on the augmented graph, with gains up to 32.6% for RotatE over the user-item data in MusIns. The proposed densification process reduces the sparsity by adding meaningful relations between the entities. All models, including the RS that are not designed to incorporate multi-relational information, benefit from the augmented data. In this setup, the RS can be thought as models that aim to predict the plausibility of an interaction between any two entities (not only user-item). Although they do not account for each particular type of
Table 4: Results for “User-item” and “Augmented” graph setups, for models operating in Euclidean (R), hyperbolic (H) and complex (C) space. Δ% shows the hit rate improvement over the user-item graph when data is augmented. All HR improvements are statistically significant (one-tailed Mann-Whitney U test, \( p \leq 0.05 \)).

| Model   | Music Instruments | Video Games | Arts, Crafts & Sewing |
|---------|-------------------|-------------|-----------------------|
|         | User-item         | Augmented   | User-item             | Augmented   | User-item             | Augmented   |
|         | DCG | HR | DCG | HR | DCG | HR | DCG | HR | DCG | HR | DCG | HR |
| BPR     | R   | .266 | 43.03 | .308 | 49.00 | 13.9 | .365 | 57.67 | .397 | 59.42 | 3.0 | .320 | 48.80 | .390 | 56.79 | 16.4 |
| HyperML | H   | .295 | 48.72 | .264 | 47.26 | -3.0 | .329 | 53.41 | .340 | 56.31 | 5.4 | .318 | 50.57 | .322 | 53.13 | 5.1 |
| CML++   | R   | .282 | 48.82 | .345 | 55.87 | 14.4 | .323 | 55.41 | .384 | 62.31 | 12.4 | .291 | 50.47 | .358 | 59.60 | 18.1 |
| NuMF    | R   | .320 | 49.88 | .347 | 55.05 | 10.4 | .419 | 62.84 | .448 | 66.93 | 6.5 | .363 | 53.35 | .420 | 61.46 | 15.2 |
| TransE  | R   | .284 | 44.86 | .346 | 54.13 | 20.7 | .387 | 58.47 | .455 | 67.82 | 16.0 | .330 | 48.49 | .421 | 60.61 | 25.0 |
| TransH  | R   | .284 | 45.05 | .342 | 54.07 | 20.0 | .390 | 58.73 | .440 | 67.51 | 14.9 | .329 | 48.22 | .422 | 61.63 | 27.8 |
| DistMul | R   | .251 | 40.58 | .310 | 49.75 | 22.6 | .346 | 54.85 | .370 | 57.54 | 4.9 | .290 | 44.85 | .321 | 48.61 | 8.4 |
| RotATOE | C   | .251 | 40.22 | .332 | 53.51 | 32.6 | .345 | 54.49 | .405 | 63.80 | 17.1 | .300 | 46.57 | .389 | 58.37 | 25.9 |
| RotRefEu | R   | .317 | 51.63 | .366 | 57.82 | 12.0 | .408 | 63.12 | .447 | 67.31 | 6.6 | .371 | 56.43 | .433 | 63.77 | 13.0 |
| RotRefHy | R   | .317 | 51.63 | .368 | 57.86 | 12.1 | .400 | 62.74 | .450 | 68.64 | 9.3 | .363 | 55.76 | .421 | 64.09 | 14.9 |
| MuRE    | R   | .337 | 52.81 | .365 | 57.99 | 9.8  | .405 | 63.11 | .462 | 68.80 | 9.0  | .372 | 56.72 | .422 | 63.95 | 12.8 |
| MuRP    | H   | .332 | 52.84 | .368 | 57.86 | 9.5  | .424 | 64.92 | .467 | 70.49 | 8.6  | .366 | 56.83 | .426 | 64.06 | 12.7 |

relation, they profit from the extended training set and this results in an enhanced generalization, which contributes to cluster users and items in a way that improves the recommendation.

In this setting, TransE and TransH show a greater relative increase in their performance (higher Δ%) compared to recommender system baselines. Since the purpose of the KG models is to cope with multi-relational data, they can exploit the augmented relations in a much better way than the RS baselines. Although the rotations in the complex plane of RotATOE offer noticeable improvements of more than 17% in all branches with the augmented graph, it is outperformed by translational approaches such as TransE.

HyperML and CML++ do not show large gains or high performances with the added relations. The reason for this is that metric learning approaches that lack relational operators are ill-posed algorithms. The reason for this is that metric learning approaches that lack relational operators are ill-posed algorithms. This was confirmed by personal correspondence with the authors of [56].

MuRP and RotRef are the best performing models also in this setup. These results highlight how heterogeneous sources of information, when leveraged through adequate tools and formulations, allow models to exploit the augmented data and achieve considerable performance gains.

Hyperbolic and Euclidean models show very competitive results for MusIns and Arts&Crafts. It has been shown that in low-dimensional setups (\( d \in [2, 32] \)) hyperbolic space offers significant improvements [11, 44, 57]. Since we operate with 64 dimensions, both models exhibit a similar representation capacity on these datasets. Nonetheless, MuRP outperforms its Euclidean counterpart in VGames for both setups. These findings are in line with the previous analysis (§4), and they emphasize the importance of choosing a suitable metric space that fits the data distribution as a powerful and efficient inductive bias. Since the amount of space in the hyperbolic representation grows exponentially compared to the Euclidean one, this model is able to accommodate entities in a better way, unfolding latent hierarchies in the graph (see §6.5).

Finally, we also demonstrate the recent developments in KGs, and their enriched representation capacities. Advanced KG methods achieve a much better performance than their predecessors and outperform RS explicitly designed for the task in both setups.

6.3 RQ3: Relation Ablation

In this section we investigate the contribution of each individual relation to the results of MuRP, which we consider the best performing model. Besides evaluating on the test split, we create a subset with the 2% of users with fewer number of interactions. They represent users affected by the cold-start problem, since they exhibit very few interactions with items. We refer to this split as “Cold Test”. The results of the ablation are presented in Table 5.

When we look at individual performance on the test set, we see that each relation brings improvements over the user-item graph (buy relation), which highlights the key role of data augmentation to boost the performance of these models. We notice that the semantic relation is the best in the “Musical Instruments” branch, whereas in the other two branches, it is the second. The relation also_bought seems to be more helpful in those cases. also_bought is mined from behavioral patterns of users with respect to complementary products that are usually bought together [53], and it plays a fundamental role in predicting user purchases [85]. However, the relations also_bought and also_view are derived from the entire dataset, and not only from the train split. These relations incorporate information between users and items on the dev/test set, posing considerable advantages over the brand, category and semantic relations.9

Finally, all relationships combined outperform the individual setups for all branches of the dataset in the test set, in line with previous results [93]. This demonstrate how KG methods can leverage heterogeneous information in an unified manner, and shows the scalability of the approach to new relation types.

9This was confirmed by personal correspondence with the authors of [56].

Cold-start Problem. We analyze how different relations affect users in cold-start settings, by looking at the 2% of users with the fewest number of interactions (Cold Test). In this case, the semantic relation brings a very pronounced boost in performance: 15.4%,
Augmenting the User-Item Graph with Textual Similarity Models

Table 5: Relation ablation for MuRP model, for different branches of the Amazon dataset. Results in bold show the best performance combination. "Cold Test" refers to the 2% of users with the smallest number of interactions. Δ% shows the improvements of each relation with respect to only using the relation buy.

| Setup | Musical Instruments | Video Games | Arts, Crafts & Sewing |
|-------|---------------------|-------------|----------------------|
|       | Test                | Cold Test   | Test                 | Cold Test   | Test                 | Cold Test   |
|       | DCG                | HR         | Δ%                  | DCG          | HR         | Δ%                     | DCG          | HR         | Δ%                     | DCG          | HR         | Δ%                     |
| buy   | .352               | 52.84      | .379               | 56.45        | .372       | 57.69               | .366         | 56.83      | .524                   | 67.12        |
| buy + brand | .351 | 54.21 | 2.6  | .418 | 58.87 | 4.3  | .425 | 64.99 | 0.1  | .404 | 61.72 | 7.0  | .371 | 58.16 | 2.3  | .533 | 68.19 | 1.6  |
| buy + category | .348 | 54.81 | 3.7  | .376 | 61.90 | 9.6  | .426 | 65.50 | 2.3  | .399 | 61.22 | 9.1  | .379 | 57.74 | 1.6  | .546 | 68.85 | 2.6  |
| buy + also_view | .352 | 55.06 | 4.2  | .449 | 62.50 | 10.5 | .456 | 68.88 | 6.1  | .416 | 62.13 | 3.7  | .385 | 59.62 | 4.9  | .545 | 69.50 | 3.5  |
| buy + also_bought | .360 | 56.06 | 6.1  | .421 | 59.27 | 5.0  | .455 | 68.90 | 6.1  | .412 | 62.50 | 8.3  | .421 | 62.92 | 10.7 | .545 | 68.49 | 2.0  |
| buy + semantic | .362 | 56.74 | 7.4  | .447 | 65.12 | 15.4 | .452 | 68.29 | 7.1  | .434 | 64.97 | 12.6 | .397 | 61.01 | 1.7  | .557 | 71.94 | 7.2  |

All relations | .355 | 56.81 | 7.3  | .460 | 67.44 | 19.5 | .467 | 70.49 | 8.6  | .416 | 62.73 | 8.7  | .426 | 64.06 | 12.7 | .541 | 69.68 | 3.8  |

Table 6: Ablation of adding item features and/or semantic similarities. Variations in Table 6 in bold show the best performing relation combination. "Cold Test" refers to the 2% of users with the smallest number of interactions. Δ% shows the improvements of each relation with respect to only using the relation buy.

12.6% and 7.2% for MusIns, VGames and Arts&Crafts respectively, much more than any other relation. This shows the effectiveness of semantic relations to densify the graph, with remarkable improvements particularly over sparse users and items. These observations are in line with previous research that has shown how reviews, when used as regularizers, are particularly helpful to alleviate cold-start problems [63]. Moreover, we notice that the performance for 'Cold Test' tends to be better than for "Test" in MusIns and Arts&Crafts. We hypothesize that this is caused by modelling users as a single point in the space. When users exhibit a large number of interactions with distinct items, it becomes more difficult to place the user embedding close to all their preferences.

6.4 RQ4: The Role of Reviews
6.4.1 Relations vs Features. We analyze the effectiveness of semantic relations to model side information extracted from textual descriptions when compared to different ways of incorporating latent features proposed in previous work. We consider the following models: CML++ utilizes the item features as an explicit regularizer, NeuMF initializes the item embeddings with the features, and Narre [13] uses TextCNN [34] to extract features from the text. We do not compare to KG approaches since they do not incorporate latent features. For CML++ and NeuMF we employ the text embeddings used to compute the semantic similarities as item features. Results for the ablation are reported in Table 6.

Compared to baselines that do not incorporate any side information, we see that relations bring a larger improvement than features for CML++ and Narre (17.1 vs 3.7 and 10.6 vs 7.4 relative improvement respectively). Although these models are not specifically designed to incorporate multi-relational information, data augmentation via semantic relations seems to be more effective than exploiting textual features. For NeuMF, the transfer learning technique of initializing the item embeddings with textual features proves to be very effective. Nonetheless, when semantic relations are combined with features, the performance is improved.

This ablation showcases the efficacy of semantic relations to model information extracted from textual data. They can be seamlessly integrated with latent features, and the improvements of the combined models demonstrate that they provide complementary information for recommendations.

6.4.2 Type of Text for Semantic Relation. We analyze which type of text is more useful for extracting features and capturing item similarities. To do so, we filter the available text for each item according to different criteria: (a) Only textual metadata, such as product name, description and categorical labels, (b) only reviews, (c) only reviews with high level of sentiment polarity, and (d) metadata + top-k longest reviews. We repeat the method described in $2.2$ to pre-process relations, and encode all text with USE. We compare CML++ and NeuMF, which are models that incorporate item features, and MuRP. In all cases we run experiments with the buy + semantic training set, in order to analyze the addition of semantic information only. Results for "MusIns" are reported in Table 7. We can see that adding features on top of relations degrade the performance of CML++ (the same behavior can be noticed in Table 6). This model integrates item features as a regularizer to correct the projection into the target embedding space. In this setup, the semantic relations are already contributing in that regard, therefore the integration of extra features seems to mislead the model. On the other hand, NeuMF has a drastic drop in performance when features are removed. In this case, the features and relations extracted from text that combine metadata with the longest reviews helps the most in the recommendations, followed by utilizing all the available reviews as input.

Finally, MuRP does not use any features and solely learns from the relations available in the graph. We again see that using metadata combined with the longest reviews is the most useful text to learn item dis/similarities. This is due to the fact that longer reviews tend to be more descriptive about the items. However, results using only metadata show competitive performance as well. In the three models, reviews with high polarity (i.e. very "positive/negative" reviews rather than "neutral" ones) do not seem to carry extra descriptive information such that it can be leveraged for similarities.
we find USE to be the most effective encoder that captures review information in a supervised manner. We compare the performance of USE to BERT \[19\], which is an adaptation of BERT with a siamese network, fine-tuned for sentence similarity. We analyze the MuRP model with the same four criteria for filtering the available text, and we compare to the setup without adding the semantic relations as well.

Results for the “MusIns” are reported in Table 8. We observe that when we use BERT as the encoder, the performance of the user-item graph extended with semantic relations is worse than using the user-item graph alone. In Figure 2 we show the cosine similarities for 50 random item embeddings. We see that for the BERT encoder, which has not been pre-trained on the hyperbolic similarity objectives, most items are very similar to each other, and this hinders the model from clustering them. This setup can be considered as an ablative experiment of the original model, where we add random semantic relations. This demonstrates the importance of the semantic information contained in the connections that we create, and how the model is able to leverage them.

Sentence-BERT shows improvements over BERT. Nevertheless, we find USE to be the most effective encoder that captures review and metadata dis/similarities. Sentence-BERT is more competent at distinguishing degrees of similarity than BERT, but Figure 2 shows that the patterns detected are alike. On the other hand, USE is able to identify a broader range of dis/similarities between items, that results in an improved performance for recommendations.

### 6.5 RQ5: Metric Space Analysis

Our experiments showed that hyperbolic methods can offer improvements over systems operating on Euclidean space or with complex numbers. Moreover, since hyperbolic space is naturally suited for embedding hierarchical structures, its self-organizing properties make it amenable to capture different types of hierarchies as a by-product of the learning process. In the resulting embeddings, the norm represents depth in the hierarchy. As explained in §3.4, each item in the Amazon dataset has a set of categorical labels that describes which categories the item belongs to. We analyze the hyperbolic and Euclidean versions of MuRP, and report the Spearman correlation between the norms of the embeddings for each category and the number of interactions of that category:

- Hyperbolic: $-0.52$
- Euclidean: $-0.06$

The correlation is moderate to high for the hyperbolic model, whereas for the Euclidean model it is non-existent. This indicates that more “general” categories (with more interactions) have a shorter norm, which is expected when embedding a hierarchy in a hyperbolic space [64], while the origin of the space has no particular meaning in the Euclidean model.

To shed light on this aspect, we reconstruct the hierarchies that the hyperbolic and Euclidean models learn. To do so, we randomly select a category embedding, and iteratively look at the closest neighbor that has less or equal norm (this would be the parent category). We report the hierarchies for “Electric Guitar Bags & Cases” and “Footswitches” from the “MusIns” dataset in Table 9.

We notice that the hyperbolic hierarchy is much more concise and precise, compared to the Euclidean one. The hyperbolic model builds different “short” hierarchies, accommodating the labels in a more spread way, whereas the Euclidean model learns a “tall” tree of categories, which results in a much more noisy arrangement.

This shows that the hyperbolic model automatically infers the hierarchy arising from the label distribution [48, 50], and offers a more interpretable space. Furthermore, the model achieves this as a by-product of the learning process with augmented relations, without being specifically trained for this purpose.

### 7 RELATED WORK

**Data Augmentation**: Data augmentation plays an important role in machine learning [39, 67], as it reduces the generalization error without affecting the effective capacity of the model [30]. In RS, augmentation techniques have been applied by extending co-purchased

---

**Tables and Figures**

**Table 7**: Comparison of models that exploit textual information extracted from different types of text. All models are trained over the user-item graph + semantic relations.

| Model          | DCG  | HR   | DCG  | HR   | DCG  | HR   |
|----------------|------|------|------|------|------|------|
| No features    | 0.346| 55.57| 0.273| 44.13| -    | -    |
| Metadata       | 0.287| 53.17| 0.310| 49.33| 0.349| 55.29|
| Reviews        | 0.334| 53.95| 0.316| 50.80| 0.354| 55.55|
| Polarity revs. | 0.321| 51.68| 0.305| 49.17| 0.344| 54.54|
| Meta + LongRevs| 0.337| 54.80| 0.320| 51.06| 0.362| 56.74|

**Table 8**: Performance of MuRP model on MusIns for different encoders, and different filtering heuristics.

| Model          | DCG  | HR   | DCG  | HR   | DCG  | HR   |
|----------------|------|------|------|------|------|------|
| User-item      | 0.332| 52.84| 0.332| 52.84| 0.332| 52.84|
| Metadata       | 0.331| 52.80| 0.334| 53.43| 0.349| 55.29|
| Reviews        | 0.325| 51.99| 0.341| 54.18| 0.354| 55.55|
| Polarity revs. | 0.310| 50.46| 0.331| 52.46| 0.344| 54.54|
| Meta + LongRevs| 0.328| 52.35| 0.339| 54.31| 0.362| 56.74|

**Figure 2**: Cosine similarity of BERT (left), S-BERT (center) and USE (right) review embeddings for 50 items.
products [85], generating adversarial pseudo-user-item interactions [82], or casting different user actions as purchases [71, 78]. Also by exploiting item side information, such as images [16], audio [46] or video [14] features. We propose a new unsupervised method to learn similarity relations between items (or users) based on semantic text models applied to textual attributes. Our work expands the strand of research that incorporates review information as regularization technique [63], and notably improves the performance for the cold-start.

**Recommenders using Text:** Previous work has mined text to use it as regularizer [31, 52], or as latent features to learn better user and item representations [13, 94]. However, Sachdeva & McAuley argue that the benefit of using reviews for recommendation is overstated, and the reported gains are only possible under a narrow set of conditions [63]. They specifically note that reviews seem to provide little benefit as features, but help more when used for regularization. The simple data augmentation method proposed in this paper offers another way to think about using textual information. The added relations improve user and item representations, like adding features can, but without needing to increase the representation size. Thus it can also be seen as providing the benefits of regularization, but without directly constraining model expressivity, as for instance dropout does.

**Knowledge Graph Recommenders:** Previous work integrating KG into RS has applied a narrow set of representational methods, favoring Euclidean translational approaches [27] (see Table 1). The experiments in this paper expand on previous analysis by including more recent KG embedding methods. Our results show that newer methods significantly benefit from the introduced data augmentation and outperform not only previous KG recommenders, but also other state-of-the-art recommendation systems.

**Hyperbolic Space:** The advantages of hyperbolic space have been argued for in a wide variety of application domains: question answering [75], machine translation [26], language modeling [20], hierarchical classification [48, 50], and taxonomy refinement [2, 41] among others. In RS, hyperbolic geometry naturally emerges in several datasets [6, 36], and hyperbolic spaces have been applied in combination with metric learning approaches [10, 79]. In this work, we expand on these studies and carry out a structural analysis of the properties of user-item graphs extended by our data augmentation, which shows that hyperbolic methods may have an important role to play for both interpretability of recommendations, and for enabling higher performance by leveraging lower representational dimensionality.

### 8 CONCLUSIONS

In this work we propose a simple unsupervised data augmentation technique that adds semantic relations to the user-item graph based on applying pre-trained language models to widely available textual attributes. This can be regarded as a data-dependent prior that introduces an effective inductive bias, without increasing the computational cost of models at inference time.

By exploring a variety of modern KG methods, we observe that recent advances, when combined with our data augmentation technique, result in state-of-the-art RS performance (RQ1, §6.1). Moreover, the proposed data augmentation improves the performance of all analyzed models, including those that are not designed to handle multi-relational information (RQ2, §6.2). Thus, the technique can be considered architecture-agnostic.

Our ablation study highlights the impact of the semantic relations particularly in cold-start settings (RQ3, §6.3). Regarding the choice of textual inputs, the study reveals that using either reviews or brief product descriptions are both effective (RQ4, §6.4). An important branch of further work here is to explore if these results generalise to denser domains, as we notice anecdotal evidence that the benefit of this data augmentation diminishes as the average degree of nodes increases.

Finally, our analysis of structural properties of the graphs, which can be extended to more datasets that exhibit a latent hyperbolic geometry [6, 10, 36], reveals that recommenders can benefit from operating in these metric spaces. In particular, we remark how hyperbolic space improves the interpretability for recommendations (RQ5, §6.5).

### ACKNOWLEDGMENTS

This work has been supported by the German Research Foundation (DFG) as part of the Research Training Group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) under grant No. GRK 1994/1 and the Klaus Tschira Foundation, Heidelberg, Germany.

### REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 2018.

[2] Rami Aly, Shantanu Acharya, Alexander Ossa, Arne Kohn, Chris Biemann, and Alexander Panchenko. Every child should have parents: A taxonomy refinement algorithm based on hyperbolic term embeddings. In *Proceedings of the 57th Annual Meeting of the ACL*, pages 4811–4817, Florence, Italy, July 2019. ACL.

[3] Ivana Balazevic, Carl Allen, and Timothy Hospedales. Multi-relational poincaré graph embeddings. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 4463–4473. NeurIPS, 2019.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 1137–1144. ACM, 2009.

[5] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[6] Pedro Cano, Oscar Celma, Markus Koppenberger, and Javier Buldu. Topology of music recommendation networks. *Chaos* (Woodbury, N.Y.), page 013107, 2006.

| Hyperbolic | Euclidean | Hyperbolic | Euclidean |
|------------|-----------|------------|-----------|
| **Norm**   | **Label** | **Norm**   | **Label** |
| 0.42       | Musical Instruments | 1.47       | Musical Instruments |
| 0.77       | Bags & Cases | 2.65       | DJ Sets |
| 0.83       | Bags & Cases | 2.70       | CD Players |
| 0.85       | Electric Guitar | 2.96       | Bags & Cases |

Table 9: Category hierarchies and norm of each category label for hyperbolic and Euclidean spaces.
Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic neural networks. [54x210]

Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. [54x236]

Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. [54x282]

Suyu Ge, Tao Qi, Chuhan Wu, Fangzhao Wu, Chuhan Wu, Fangzhao Wu, X. Xie, and Y. Huang. Helpfulness-aware review based neural recommendation. [54x289]

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. [54x308]

Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Antonia F likeness. Collaborative metric learning. [54x324]

Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. Improving sequential recommendation with knowledge-enhanced memory networks. [54x342]

Guoliang Ji, Shizhu He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. Neural attentional rating regression with review-level explanations. [54x359]

Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. Neural collaborative filtering. In Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys ’17, page 161–169, New York, NY, USA, 2017. ACM.
[92] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 2019.

[93] Yongfeng Zhang, Qingyao Ai, Xu Chen, and Pengfei Wang. Learning over knowledge-base embeddings for recommendation. *CoRR*, abs/1803.06540, 2018.

[94] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 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*, WSDM ’17, page 425–434, New York, NY, USA, 2017. ACM.