TransRev: Modeling Reviews as Translations from Users to Items

Alberto García-Durán[1(✉)], Roberto González[2], Daniel Oñoro-Rubio[2], Mathias Niepert[2], and Hui Li[3]

[1] EPFL, Lausanne, Switzerland
alberto.duran@epfl.ch
[2] NEC Labs Europe, Heidelberg, Germany
{roberto.gonzalez,daniel.onoro,mathias.niepert}@neclab.eu
[3] Xiamen University, Xiamen, China
huili.xmu@gmail.com

Abstract. The text of a review expresses the sentiment a customer has towards a particular product. This is exploited in sentiment analysis where machine learning models are used to predict the review score from the text of the review. Furthermore, the products costumers have purchased in the past are indicative of the products they will purchase in the future. This is what recommender systems exploit by learning models from purchase information to predict the items a customer might be interested in. The underlying structure of this problem setting is a bipartite graph, wherein customer nodes are connected to product nodes via 'review' links. This is reminiscent of knowledge bases, with 'review' links replacing relation types. We propose TransRev, an approach to the product recommendation problem that integrates ideas from recommender systems, sentiment analysis, and multi-relational learning into a joint learning objective.

TransRev learns vector representations for users, items, and reviews. The embedding of a review is learned such that (a) it performs well as input feature of a regression model for sentiment prediction; and (b) it always translates the reviewer embedding to the embedding of the reviewed item. This is reminiscent of TransE [5], a popular embedding method for link prediction in knowledge bases. This allows TransRev to approximate a review embedding at test time as the difference of the embedding of each item and the user embedding. The approximated review embedding is then used with the regression model to predict the review score for each item. TransRev outperforms state of the art recommender systems on a large number of benchmark data sets. Moreover, it is able to retrieve, for each user and item, the review text from the training set whose embedding is most similar to the approximated review embedding.

Keywords: Recommender systems · Knowledge graphs · Sentiment analysis

J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12035, pp. 234–248, 2020.
https://doi.org/10.1007/978-3-030-45439-5_16

1 Introduction

Online retail is a growing market with sales accounting for $394.9 billion or 11.7% of total US retail sales in 2016 [35]. In the same year, e-commerce sales accounted for 41.6% of all retail sales growth [15]. For some entertainment products such as movies, books, and music, online retailers have long outperformed traditional in-store retailers. One of the driving forces of this success is the ability of online retailers to collect purchase histories of customers, online shopping behavior, and reviews of products for a very large number of users. This data is driving several machine learning applications in online retail, of which personalized recommendation is the most important one. With recommender systems online retailers can provide personalized product recommendations and anticipate purchasing behavior.

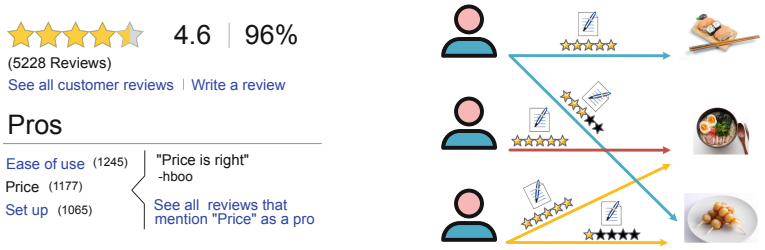

Fig. 1. (Left) A typical product summary with review score and 'Pros'. Image taken from www.bestbuy.com. (Right) A small bipartite graph modeling customers (users), products (items), product reviews, and review scores.

In addition, the availability of product reviews allows users to make more informed purchasing choices and companies to analyze costumer sentiment towards their products. The latter was coined sentiment analysis and is concerned with machine learning approaches that map written text to scores. Nevertheless, even the best sentiment analysis methods cannot help in determining which *new* products a costumer might be interested in. The obvious reason is that costumer reviews are not available for products they have not purchased yet.

In recent years the availability of large corpora of product reviews has driven text-based research in the recommender system community (e.g. [3,19,21]). Some of these novel methods extend latent factor models to leverage review text by employing an explicit mapping from text to either user or item factors. At prediction time, these models predict product ratings based on some operation (typically the dot product) applied to the user and product representations. Sentiment analysis, however, is usually applied to some representation (e.g. bag-of-words) of review text but in a recommender system scenario the review is not available at prediction time.

With this paper we propose TransRev, a method that combines a personalized recommendation learning objective with a sentiment analysis objective into a joint learning objective. TransRev learns vector representations for

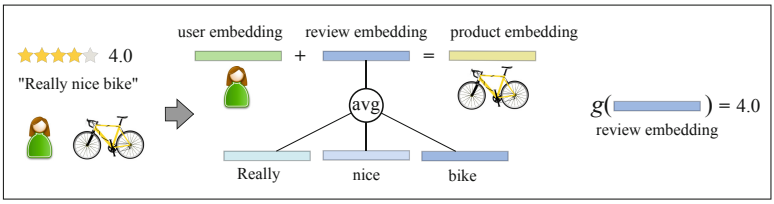

Fig. 2. At training time, a function's parameters are learned to compute the review embedding from the word token embeddings such that the embedding of the user translated by the review embedding is similar to the product embedding. At the same time, a regression model g is trained to perform well on predicting ratings.

users, items, and reviews jointly. The crucial advantage of TransRev is that the review embedding is learned such that it corresponds to a translation that moves the embedding of the reviewing user to the embedding of the item the review is about. This allows TransRev to approximate a review embedding at test time as the difference of the item and user embedding despite the absence of a review from the user for that item. The approximated review embedding is then used in the sentiment analysis model to predict the review score. Moreover, the approximated review embedding can be used to retrieve reviews in the training set deemed most similar by a distance measure in the embedding space. These retrieved reviews could be used for several purposes. For instance, such reviews could be provided to users as a starting point for a review, lowering the barrier to writing reviews.

2 TransRev: Modeling Reviews as Translations in Vector Space

We address the problem of learning prediction models for the product recommendation problem. A small example of the input data typical to such a machine learning system is depicted in Fig. 1. This reminds of knowledge bases, with 'reviews' replacing relation types. Two nodes in a knowledge base may be joined by a number of links, each representing one relation type from a small vocabulary. Here, if two nodes are connected they are linked by one single edge type, in which case it is represented by a number of words from a (very) large vocabulary.

There are a set of users $\mathbf{U}$, a set of items $\mathbf{I}$, and a set of reviews $\mathbf{R}$. Each $\mathtt{rev}_{(\mathtt{u},\mathtt{i})} \in \mathbf{R}$ represents a review written by user $\mathtt{u}$ for item $\mathtt{i}$. Hence, $\mathtt{rev}_{(\mathtt{u},\mathtt{i})} = [\mathtt{t}_1, \cdots, \mathtt{t}_n]$, that is, each review is a sequence of n tokens. In the following we refer to $(\mathtt{u}, \mathtt{rev}_{(\mathtt{u},\mathtt{i})}, \mathtt{i})$ as a *triple*. Each such triple is associated with the review score $\mathtt{r}_{(\mathtt{u},\mathtt{i})}$ given by the user $\mathtt{u}$ to item $\mathtt{i}$.

TransRev embeds all users, items and reviews into a latent space where the embedding of a user plus the embedding of the review is learned to be close to the embedding of the reviewed item. It simultaneously learns a regression model to predict the rating given a review text. This is illustrated in Fig. 2.

At prediction time, reviews are not available, but the modeling assumption of TransRev allows to predict the review embedding by taking the difference of the embedding of the item and user. Then this approximation is used as input feature of the regression model to perform rating prediction—see Fig. 3.

TransRev embeds all nodes and reviews into a latent space $\mathbb{R}^k$ (k is a model hyperparameter). The review embeddings are computed by applying a learnable function f to the token sequence of the review

$$\mathbf{h}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}} = f(\mathtt{rev}_{(\mathtt{u},\mathtt{i})}).$$

The function f can be parameterized (typically with a neural network such as a recursive or convolutional neural network) but it can also be a simple parameter-free aggregation function that computes, for instance, the element-wise average or maximum of the token embeddings.

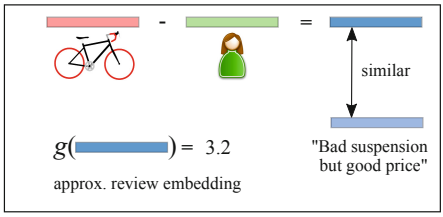

Fig. 3. At test time, the review embedding is approximated as the difference between the product and user embeddings. The approximated review embedding is used to predict the rating and to retrieve similar reviews.

We propose and evaluate a simple instance of f where the review embedding $\mathbf{h}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}}$ is the average of the embeddings of the tokens occurring in the review. More formally,

$$\mathbf{h}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}} = f(\mathtt{rev}_{(\mathtt{u},\mathtt{i})}) = \frac{1}{|\mathtt{rev}_{(\mathtt{u},\mathtt{i})}|} \sum_{\mathtt{t} \in \mathtt{rev}_{(\mathtt{u},\mathtt{i})}} \mathbf{v}_{\mathtt{t}} + \mathbf{h}_0, \tag{1}$$

where $\mathbf{v}_{\mathtt{t}}$ is the embedding associated with token $\mathtt{t}$ and $\mathbf{h}_0$ is a review bias which is common to all reviews and takes values in $\mathbb{R}^k$. The review bias is of importance since there are some reviews all of whose tokens are not in the training vocabulary. In these cases we have $\mathbf{h}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}} = \mathbf{h}_0$.

The learning of the item, review, and user embeddings is determined by two learning objectives. The first objective guides the joint learning of the parameters of the regression model and the review embeddings such that the regression model performs well at review score prediction

$$\min \mathcal{L}_1 = \min \sum_{((\mathtt{u},\mathtt{rev}_{(\mathtt{u},\mathtt{i})},\mathtt{i}),\mathtt{r}_{(\mathtt{u},\mathtt{i})}) \in S} \left(g(\mathbf{h}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}}) - \mathtt{r}_{(\mathtt{u},\mathtt{i})}\right)^2, \tag{2}$$

where S is the set of training triples and their associated ratings, and g is a learnable regression function $\mathbb{R}^k \to \mathbb{R}$ that is applied to the representation of the review $\mathbf{h}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}}$.

While g can be an arbitrary complex function, the instance of g used in this work is as follows

$$g(\mathbf{h}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}}) = \sigma(\mathbf{h}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}})\mathbf{w}^T + \mathtt{b}_{(\mathtt{u},\mathtt{i})}, \tag{3}$$

where $\mathbf{w}$ are the learnable weights of the linear regressor, σ is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, and $\mathtt{b}_{(\mathtt{u},\mathtt{i})}$ is the shortcut we use to refer to the sum of the bias terms, namely the user, item and overall bias: $\mathtt{b}_{(\mathtt{u},\mathtt{i})} = \mathtt{b}_{\mathtt{u}} + \mathtt{b}_{\mathtt{i}} + \mathtt{b}_0$. Later we motivate the application of the sigmoid function to the review embedding.

Of course, in a real-world scenario a recommender system makes rating predictions on items that users have *not rated yet* and, consequently, reviews are not available for those items. The application of the regression model of Eq. (3) to new examples, therefore, is not possible at test time. Our second learning procedure aims at overcoming this limitation by leveraging ideas from embedding-based knowledge base completion methods. We want to be able to approximate a review embedding at test time such that this review embedding can be used in conjunction with the learned regression model. Hence, in addition to the learning objective (2), we introduce a second objective that forces the embedding of a review to be close to the difference between the item and user embeddings. This translation-based modeling assumption is followed in TransE [5] and several other knowledge base completion methods [10,13]. We include a second term in the objective that drives the distance between (a) the user embedding translated by the review embedding and (b) the embedding of the item to be small

$$\min \mathcal{L}_2 = \min \sum_{((\mathtt{u},\mathtt{rev}_{(\mathtt{u},\mathtt{i})},\mathtt{i}),\mathtt{r}_{(\mathtt{u},\mathtt{i})})\in S} ||\mathbf{e}_{\mathtt{u}} + \mathbf{h}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}} - \mathbf{e}_{\mathtt{i}}||_2, \tag{4}$$

where $\mathbf{e}_{\mathtt{u}}$ and $\mathbf{e}_{\mathtt{i}}$ are the embeddings of the user and item, respectively. In the knowledge base embedding literature (cf. [5]) it is common the representations are learned via a margin-based loss, where the embeddings are updated if the score (the negative distance) of a positive triple (e.g. $(\mathtt{Berlin}, \mathtt{located\_in}, \mathtt{Germany})$) is not larger than the score of a negative triple (e.g. $(\mathtt{Berlin}, \mathtt{located\_in}, \mathtt{Portugal})$) plus a margin. Note that this type of learning is required to avoid trivial solutions. The minimization problem of Eq. (4) can easily be solved by setting $\mathbf{e}_{\mathtt{u}} = \mathbf{h}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}} = \mathbf{e}_{\mathtt{i}} = \mathbf{0}\ \forall \mathtt{u}, \mathtt{i}$. However, this kind of trivial solutions is avoided by jointly optimizing Eqs. (2) and (4), since a degenerate solution like the aforementioned one would lead to a high error with respect to the regression objective (Eq. (2)). The overall objective can now be written as

$$\min_{\Theta} \mathcal{L} = \min_{\Theta} (\mathcal{L}_1 + \lambda \mathcal{L}_2 + \mu ||\Theta||_2), \tag{5}$$

where λ is a term that weights the approximation loss due to the modeling assumption formalized in Eq. (4). In our model, Θ corresponds to the parameters $\mathbf{w}$, $\mathbf{e}$, $\mathbf{v}$, $\mathbf{h}_0 \in \mathbb{R}^k$ and the bias terms $\mathtt{b}$.

At test time, we can now approximate review embeddings of $(\mathtt{u}, \mathtt{i})$ pairs *not seen* during training by computing

$$\boxed{\hat{\mathbf{h}}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}} = \mathbf{e}_{\mathtt{i}} - \mathbf{e}_{\mathtt{u}}.} \tag{6}$$

With the trained regression model g we can make rating predictions $\hat{\mathtt{r}}_{(\mathtt{u},\mathtt{i})}$ for *unseen* $(\mathtt{u},\mathtt{i})$ pairs by computing

$$\hat{\mathtt{r}}_{(\mathtt{u},\mathtt{i})} = g(\hat{\mathbf{h}}_{\mathtt{rev}_{\mathtt{u},\mathtt{i}}}). \tag{7}$$

Contrary to training, now the regression model g is applied to $\hat{\mathbf{h}}_{\mathtt{rev}_{\mathtt{u},\mathtt{i}}}$, instead of $\mathbf{h}_{\mathtt{rev}_{\mathtt{u},\mathtt{i}}}$, which is not available at test time. The sigmoid function of the regression function g adds a non-linear interaction between the user and item representation. Without such activation function, the model would consist of a linear combination of bias terms and the (ranking of) served recommendations would be identical to all users.

All parameters of the parts of the objective are jointly learned with stochastic gradient descent. More details regarding the parameter learning are contained in the experimental section.

2.1 On the Choice of TransE as Modeling Assumption

The choice of TransE as underlying modeling assumption to this recommendation problem is not arbitrary. Given the user and item embeddings, and without further constraints, it allows to distinctively compute the approximate review embedding via Eq. (6). Another popular knowledge graph embedding method is DistMult [16]. In applying such modeling assumption to this problem one would obtain the approximate review embedding by solving the following optimization problem: $\hat{\mathbf{h}}_{\mathtt{rev}_{(\mathtt{u},\mathtt{i})}} = \max_{\mathbf{h}}(\mathbf{e}_{\mathtt{i}} \circ \mathbf{e}_{\mathtt{u}})\mathbf{h}$, where $\circ$ is the element-wise multiplication. The solution to that problem would be any vector with infinite norm. Therefore, one should impose constraints in the norm of the embeddings to obtain a non-trivial solution. However, previous work [11] shows that such constraint harms performance. Similarly, most of the knowledge graph embedding methods would require to impose constraints in the norm of the embeddings. The translation modeling assumption of TransE facilitates the approximation of the review embedding without additional constraints, while its performance is on par with, if not better, than most of all other translation-based knowledge graph embedding methods [11].

3 Related Work

There are three lines of research related to our work: knowledge graph completion, recommender systems and sentiment analysis.

The first research theme related to TransRev is knowledge graph completion. In the last years, many embedding-based methods have been proposed to infer missing relations in knowledge graphs based on a function that computes a likelihood score based on the embeddings of entities and relation types. Due to its simplicity and good performance, there is a large body of work on translation-based scoring functions [5,13]. [14] propose an approach to large-scale sequential sales prediction that embeds items into a transition space where user

embeddings are modeled as translation vectors operating on item sequences. The associated optimization problem is formulated as a sequential Bayesian ranking problem [28]. To the best of our knowledge, [14] is the first work in leveraging ideas from knowledge graph completion methods for recommender system. Whereas TransRev addresses the problem of rating prediction by incorporating review text, [14] addresses the different problem of sequential recommendation. Therefore the experimental comparison to that work is not possible. In TransRev the review embedding translates the user embedding to the product embedding. In [14], the user embedding translates a product embedding to the embedding of the next purchased product. Moreover, TransRev gets rid of the margin-based loss (and consequently of the negative sampling) due to the joint optimization of Eqs. (2) and (4), whereas [14] is formalized as a ranking problem in a similar way to [5]. Subsequently, there has been additional work on translation-based models in recommender systems [25,33]. However, these works cannot incorporate users' feedback other than ratings into the learning, which has been shown to boost performance [21].

There is an extensive body of work on recommender systems [1,6,29]. Singular Value Decomposition (SVD) [17] computes the review score prediction as the dot product between the item embeddings and the user embeddings plus some learnable bias terms. Due to its simplicity and performance on numerous data sets—including winning solution to the Netflix prize—it is still one of the most used methods for product recommendations. Most of the previous research that explored the utility of review text for rating prediction can be classified into two categories.

Semi-supervised Approaches. HFT [21] was one of the first methods combining a supervised learning objective to predict ratings with an unsupervised learning objective (e.g. latent Dirichlet allocation) for text content to regularize the parameters of the supervised model. The idea of combining two learning objectives has been explored in several additional approaches [3,9,19]. The methods differ in the unsupervised objectives, some of which are tailored to a specific domain. For example, JMARS [9] outperforms HFT on a movie recommendation data set but it is outperformed by HFT on data sets similar to those used in our work [36].

Supervised Approaches. Methods that fall into this category such as [31,32] learn latent representations of users and items from the text content so as to perform well at rating prediction. The learning of the latent representations is done via a deep architecture. The approaches differences lie mainly in the neural architectures they employ.

There is one crucial difference between the aforementioned methods and TransRev. TransRev predicts the review score based on an approximation of the review embedding computed at test time. Moreover, since TransRev is able to approximate a review embedding, we can use this embedding to retrieve reviews in the training set deemed most similar by a distance metric in the embedding space.

Similar to sentiment analysis methods, TransRev trains a regression model that predicts the review rating from the review text. Contrary to the typical setting in which sentiment analysis methods operate, however, review text is not available at prediction time in the recommender system setting. Consequently, the application of sentiment analysis to recommender systems is not directly possible. In the simplest case, a sentiment analysis method is a linear regressor applied to a text embedding (Eq. (3)).

4 Experimental Setup

We conduct several experiments to empirically compare TransRev to state of the art methods for product recommendation. Moreover, we provide some qualitative results on retrieving training reviews most similar to the approximated reviews at test time.

4.1 Data Sets

We evaluate the various methods on data sets from the Amazon Product Data[1], which has been extensively used in previous works [21–23]. The data set consists of reviews and product metadata from Amazon from May 1996 to July 2014. We focus on the 5-core versions (which contain at least 5 reviews for each user and item) of those data sets. There are 24 product categories from which we have randomly picked 18. As all previously mentioned works, we treat each of these resulting 18 data sets independently in our experiments. Ratings in all benchmark data sets are integer values between 1 and 5. As in previous work, we randomly sample 80% of the reviews as training, 10% as validation, and 10% as test data. We remove reviews from the validation and test splits if they involve either a product or a user that is not part of the training data.

4.2 Review Text Preprocessing

We follow the same preprocessing steps for each data set. First, we lowercase the review texts and apply the regular expression "$\backslash w+$" to tokenize the text data, discarding those words that appear in less than 0.1% of the reviews of the data set under consideration. For all the Amazon data sets, both full reviews and short summaries (rarely having more than 30 words) are available. Since classifying short documents into their sentiment is less challenging than doing the same for longer text [4], we have used the reviews summaries for our work. We truncate these reviews to the first 200 words. For lack of space we cannot include statistics of the preprocessed data sets.

[1] http://jmcauley.ucsd.edu/data/amazon.

4.3 Baselines

We compare to the following methods: a SVD matrix factorization; HFT, which has not often been benchmarked in previous works; and DeepCoNN [38], which learns user and item representations from reviews via convolutional neural networks. We also include MPCN [34] (which stands for multi-pointer co-attention networks) in the comparison, however, as indicated in previous work [8] MPCN is a non-reproducible work[2]. Therefore, we simply copy numbers from [34], since they used the same data sets as the ones used in this work. Additionally, we also include performance for TransNets (T-Nets) [7], whose numbers are also copied from [34]. T-Nets is similar to TransRev in that it also infers review latent representations from user and item representations. Different to TransRev, it does not have any underlying graph-based modeling assumption among users, items and reviews.

Table 1. Performance comparison (MSE) on 18 datasets. The asterisk $*$ indicates the macro MSE across all the Amazon data sets.

	HFT	SVD	DeepCoNN	T-Nets	MPCN	TransRev
Amazon Instant Video	0.888	0.904	0.943	1.007	0.997	**0.884**
Automotive	0.862	0.857	**0.853**	0.946	0.861	0.855
Baby	1.104	1.108	1.154	1.338	1.304	**1.100**
Cds and Vinyl	**0.854**	0.863	0.888	1.010	1.005	**0.854**
Grocery and Gourmet Food	0.961	0.964	0.973	1.129	1.125	**0.957**
Health and personal care	1.014	1.016	1.081	1.249	1.238	**1.011**
Kindle Store	**0.593**	0.607	0.648	0.797	0.775	0.599
Musical Instruments	0.692	0.694	0.723	1.100	0.923	**0.690**
Office Products	0.727	0.727	0.738	0.840	0.779	**0.724**
Patio, Lawn and Garden	0.956	0.950	1.070	1.123	1.011	**0.941**
Pet Supplies	1.194	1.198	1.281	1.346	1.328	**1.191**
Tools and Home Improvement	0.884	0.884	0.946	1.122	1.096	**0.879**
Toys and Games	**0.784**	0.788	0.851	0.974	0.973	**0.784**
Beauty	1.165	1.168	1.184	1.404	1.387	**1.158**
Digital Music	0.793	0.797	0.835	1.004	0.970	**0.782**
Video Games	1.086	1.093	1.133	1.276	1.257	**1.082**
Sports and Outdoors	0.824	0.828	0.882	0.994	0.980	**0.823**
Cell Phones and Accessories	1.285	1.290	1.365	1.431	1.413	**1.279**
	0.926*	0.930*	0.969*	1.116*	1.079*	**0.921***

[2] A work is considered to be reproducible if a working version of the source code is available, and at least one dataset used in the original paper is available.

4.4 Parameter Setting

We set the dimension k of the embedding space to 16 for all methods. We evaluated the robustness of TransRev to changes in Sect. 4.6. Alternatively, one could use off-the-shelf word embeddings (e.g. word2vec [24] or ELMO [26]), but this would require to assume the existence of a large collection of text for effectively learning good word representations in an unsupervised manner. However, such a corpus may not be available for some low-resource languages or domain-specific use cases. For HFT we used the original implementation of the authors[3] and validated the trade-off term from the values $[0.001, 0.01, 0.1, 1, 10, 50]$. For TransRev we validated λ among the values $[0.05, 0.1, 0.25, 0.5, 1]$ and the learning rate of the optimizer and regularization term (μ in our model) from the values $[0.001, 0.005, 0.01, 0.05, 0.1]$ and $[0.00001, 0.00005, 0.0001, 0.0005, 0.001]$, respectively. TransRev's parameters were randomly initialized [12] and learned with vanilla stochastic gradient descent. A single learning iteration performs SGD with all review triples in the training data and their associated ratings. For TransRev we used a batch size of 64. We ran TransRev for a maximum of 500 epochs and validated every 10 epochs. For SVD we used the Python package Surprise[4], and chose the learning rate and regularization term from the same range of values. Parameters for HFT were learned with L-BFGS, which was run for 2,500 learning iterations and validated every 50 iterations. For DeepCoNN the original authors' code is not available and we used a third-party implementation[5]. We applied the default hyperparameters values for dropout and L2 regularization and used the same embedding dimension as for all other methods. All methods are validated according to the Mean Squared Error (MSE).

4.5 Results

The experimental results are listed in Table 1 where the best performance is in bold font. TransRev achieves the best performance on all data sets with the exception of the Kindle Store and Automotive categories. Surprisingly, HFT is more competitive than more recent approaches that also take advantage of review text. Most of these recent approaches do not include HFT in their baselines. TransRev is competitive with and often outperforms HFT on the benchmark data sets under consideration. To quantify that the rating predictions made by HFT and TransRev are significantly different we have computed the dependent t-test for paired samples and for all data sets where TransRev outperforms HFT. The p-value is always smaller than 0.01.

It is remarkable the low performance of DeepCoNN, MPCN and T-Nets in almost all datasets. This is in line with the findings reported in very recent work [8], where authors' analysis reveals that deep recommender models are systematically outperformed by simple heuristic recommender methods. These results only confirm the existing problem reported in [8].

[3] http://cseweb.ucsd.edu/jmcauley/code/code_RecSys13.tar.gz.
[4] http://surpriselib.com/.
[5] https://github.com/chenchongthu/DeepCoNN.

4.6 Hyperparameters

We randomly selected the 4 data sets *Baby, Digital Music, Office* and *Tools&Home Improvement* from the Amazon data and evaluated different values of k for user, item and word embedding sizes. We increase k from 4 to 64 and always validate all hyperparameters, including the regularization term. Table 2 list the MSE scores. We only observe small differences in the corresponding model's performances. This observation is in line with [21].

Table 2. Sensitivity to latent dimension.

k	Baby	Digital Music	Office products	Tools& Home Improv.
4	1.100	0.782	0.724	0.880
8	1.100	0.782	0.723	0.878
16	1.100	0.782	0.724	0.879
32	1.102	0.785	0.722	0.888
64	1.099	0.787	0.726	0.888

For most of the data sets the validated weighting term λ takes the value of either 0.1 or 0.25. This seems to indicate that the regression objective is more important than the modeling assumption in our task, as it directly relates to the goal of the task. The regularization term is of crucial importance to obtain good performance and largely varies across data sets, as their statistics also largely differ across data sets.

4.7 Visualization of the Word Embeddings

Review embeddings, which are learned from word embeddings, are learned to be good predictors of user ratings. As a consequence the learned word embeddings are correlated with the ratings. To visualize the correlation between words and ratings we proceed as follows. First, we assign a score to each word that is computed by taking the average rating of the reviews that contain the word. Second, we compute a 2-dimensional representation of the words by applying t-SNE [20] to the 16-dimensional word embeddings learned by TransRev. Figure 4 depicts these 2-dimensional word embedding vectors learned for the Amazon *Beauty* data set. The corresponding rating scores are indicated by the color.

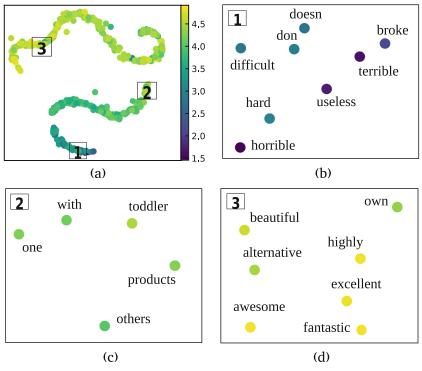

Fig. 4. (a) Two-dimensional t-SNE representations of the word embeddings learned by TransRev for the *Beauty* data set. The color bar represents the average rating of the reviews where each word appears. (b), (c) and (d) depict regions of the embedding space where negative, neutral and positive words are clustered, respectively. (Color figure online)

The clusters we discovered in Fig. 4 are interpretable. They are meaningful with respect to the score, observing that the upper right cluster is mostly made up of words with negative connotations (e.g. horrible, useless...), the lower left one contains neutral words (e.g. with, products...) and the lower right one contains words with positive connotations (e.g. awesome, excellent...).

4.8 Suggesting Reviews to Users

One of the characteristics of TransRev is its ability to approximate the review representation at prediction time. This approximation is used to make a rating prediction, but it can also be used to propose a tentative review on which the user can elaborate on. This is related to a number of approaches [18,27,37] on explainable recommendations. We compute the Euclidean distance between the approximated review embedding $\hat{\mathbf{h}}_{\mathtt{rev}_{(u,i)}}$ and all review embeddings $\mathbf{h}_{\mathtt{rev}_{(u,i)}}$ from the training set. We then retrieve the review text with the most similar review embedding. We investigate the quality of the tentative reviews that TransRev retrieves for the *Beauty* and *Digital Music* data sets. The example reviews listed in Table 3 show that while the overall sentiment is correct in most cases, we can also observe the following shortcomings: (a) The function f chosen in our work is invariant to word ordering and, therefore, cannot learn that bigrams such as "not good" have a negative meaning. (b) Despite matching the overall sentiment, the actual and retrieved review can refer to different aspects of the product (for example, "it clumps" and "gives me headaches"). Related work [37] extracts aspects from reviews by applying a number of grammatical and morphological analysis tools. These aspects are used later on to explain why the model suspects that a user might be interested in a certain product. We think this type of explanation is complementary to ours, and might inspire future work. (c) Reviews can be specific to a single product. A straightforward improvement could consist of retrieving only existing reviews for the specific product under consideration.

Table 3. Reviews retrieved from the *Beauty* (upper) and *Digital Music* (lower) data sets. In parenthesis the ratings associated to the reviews.

Actual test review	Closest training review in embedding space
skin improved (5)	makes your face feel refreshed (5)
love it (5)	you'll notice the difference (5)
best soap ever (5)	I'll never change it (5)
it clumps (2)	gives me headaches (1)
smells like bug repellent (3)	pantene give it up (2)
fake fake fake do not buy (1)	seems to be harsh on my skin (2)
saved my skin (5)	not good quality (2)
another great release from saliva (5)	can't say enough good things about this cd (5)
a great collection (5)	definitive collection (5)
sound nice (3)	not his best nor his worst (4)
a complete massacre of an album (2)	some great songs but overall a disappointment (3)
the very worst best of ever (1)	overall a pretty big disappointment (2)
what a boring moment (1)	overrated but still alright (3)
great cd (5)	a brilliant van halen debut album (5)

We believe that more sophisticated sentence and paragraph representations might lead to better results in the review retrieval task. As discussed, a promising line of research has to do with learning representations for reviews that are aspect-specific (*e.g.* "ease of use" or "price").

5 Conclusion

TransRev is a novel approach for product recommendation combining ideas from knowledge graph embedding methods, recommender systems and sentiment analysis. TransRev achieves state of the art performance on the data sets under consideration while having fewer (hyper)parameters than more recent works.

Most importantly, one main characteristic of TransRev is its ability to approximate the review representation during inference. This approximated representation can be used to retrieve reviews in the training set that are similar with respect to the overall sentiment towards the product. Such reviews can be dispatched to users as a starting point for a review, and thus lowering the barrier to writing new reviews. Given the known influence of product reviews in the purchasing choices of the users [2,30], we think that recommender systems will benefit from such mechanism.

Acknowledgements. The research leading to these results has received funding from the European Union's Horizon 2020 innovation action programme under grant agreement No 786741 – SMOOTH project. This publication reflects only the author's views and the European Community is not liable for any use that may be made of the information contained herein.

References

1. Allen, R.B.: User models: theory, method, and practice. Int. J. Man Mach. Stud. **32**(5), 511–543 (1990)
2. Amazon. https://www.amzinsight.com/amazon-product-review-importance/
3. Bao, Y., Fang, H., Zhang, J.: TopicMF: simultaneously exploiting ratings and reviews for recommendation. In: AAAI, pp. 2–8 (2014)
4. Bermingham, A., Smeaton, A.F.: Classifying sentiment in microblogs: is brevity an advantage? In: CIKM, pp. 1833–1836 (2010)
5. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS, pp. 2787–2795 (2013)
6. Breese, J.S., Heckerman, D., Kadie, C.M.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI, pp. 43–52 (1998)
7. Catherine, R., Cohen, W.W.: TransNets: learning to transform for recommendation. In: RecSys, pp. 288–296 (2017)
8. Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In: RecSys (2019)
9. Diao, Q., Qiu, M., Wu, C., Smola, A.J., Jiang, J., Wang, C.: Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In: KDD, pp. 193–202 (2014)
10. García-Durán, A., Bordes, A., Usunier, N.: Composing relationships with translations. In: EMNLP, pp. 286–290. The Association for Computational Linguistics (2015)
11. García-Durán, A., Bordes, A., Usunier, N., Grandvalet, Y.: Combining two and three-way embedding models for link prediction in knowledge bases. J. Artif. Intell. Res. **55**, 715–742 (2016)

12. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS. JMLR Proceedings, vol. 9, pp. 249–256 (2010)
13. Guu, K., Miller, J., Liang, P.: Traversing knowledge graphs in vector space. In: EMNLP, pp. 318–327. The Association for Computational Linguistics (2015)
14. He, R., Kang, W., McAuley, J.: Translation-based recommendation. In: RecSys, pp. 161–169 (2017)
15. Joe, M.: https://www.matrixmarketinggroup.com/product-taxonomy-ecommerce-sales/
16. Kadlec, R., Bajgar, O., Kleindienst, J.: Knowledge base completion: baselines strike back. arXiv preprint arXiv:1705.10744 (2017)
17. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Comput. **42**(8), 30–37 (2009)
18. Lawlor, A., Muhammad, K., Rafter, R., Smyth, B.: Opinionated explanations for recommendation systems. In: Bramer, M., Petridis, M. (eds.) Research and Development in Intelligent Systems XXXII, pp. 331–344. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25032-8_25
19. Ling, G., Lyu, M.R., King, I.: Ratings meet reviews, a combined approach to recommend. In: RecSys, pp. 105–112 (2014)
20. Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)
21. McAuley, J.J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: RecSys, pp. 165–172 (2013)
22. McAuley, J.J., Pandey, R., Leskovec, J.: Inferring networks of substitutable and complementary products. In: KDD, pp. 785–794 (2015)
23. McAuley, J.J., Targett, C., Shi, Q., van den Hengel, A.: Image-based recommendations on styles and substitutes. In: SIGIR, pp. 43–52 (2015)
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
25. Palumbo, E., Rizzo, G., Troncy, R., Baralis, E., Osella, M., Ferro, E.: An empirical comparison of knowledge graph embeddings for item recommendation. In: DL4KGS@ ESWC, pp. 14–20 (2018)
26. Peters, M.E., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
27. Qureshi, M.A., Greene, D.: *Lit@EVE*: explainable recommendation based on wikipedia concept vectors. In: Altun, Y., et al. (eds.) ECML PKDD 2017. LNCS (LNAI), vol. 10536, pp. 409–413. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71273-4_41
28. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: WWW, pp. 811–820 (2010)
29. Rennie, J.D.M., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: ICML. ACM International Conference Proceeding Series, vol. 119, pp. 713–719 (2005)
30. Saleh, K.: https://www.invespcro.com/blog/the-importance-of-online-customer-reviews-infographic/
31. Seo, S., Huang, J., Yang, H., Liu, Y.: Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In: RecSys, pp. 297–305 (2017)

32. Seo, S., Huang, J., Yang, H., Liu, Y.: Representation learning of users and items for review rating prediction using attention-based convolutional neural network. In: 3rd International Workshop on Machine Learning Methods for Recommender Systems (MLRec) (2017)
33. Tay, Y., Anh Tuan, L., Hui, S.C.: Latent relational metric learning via memory-based attention for collaborative ranking. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, pp. 729–739. International World Wide Web Conferences Steering Committee (2018)
34. Tay, Y., Tuan, L.A., Hui, S.C.: Multi-pointer co-attention networks for recommendation. In: KDD (2018)
35. Wallace, T.: https://www.bigcommerce.com/blog/ecommerce-sales-funnel/
36. Wu, C., Beutel, A., Ahmed, A., Smola, A.J.: Explaining reviews and ratings with PACO: poisson additive co-clustering. In: WWW (Companion Volume), pp. 127–128 (2016)
37. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: SIGIR, pp. 83–92 (2014)
38. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: WSDM, pp. 425–434 (2017)