Knowledge-Based Systems 228 (2021) 107282

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Learning product representations for generating reviews for cold products

Fatemeh Pourgholamali [a,c], Mohsen Kahani [b], Zeinab Noorian [c], Ebrahim Bagheri [c,*]

[a] Department of Computer Engineering, Vali-e-Asr University of Rafsanjan, Iran
[b] Web Technology Lab, Department of Computer Engineering, Ferdowsi University of Mashhad, Iran
[c] Ryerson University, Toronto, Canada

ARTICLE INFO

Article history:
Received 18 August 2020
Received in revised form 1 April 2021
Accepted 2 July 2021
Available online 8 July 2021

Keywords:
Recommender Systems
Ecommerce
Cold products

ABSTRACT

Existing work in the literature have shown that the number and quality of product ratings and reviews have a direct correlation with the product purchase rates in online e-commerce portals. However, the majority of the products on e-commerce portals do not have any ratings or reviews and are known as cold products (∼90% of products on Amazon are cold). As such, there has been growing interest in generating reviews for cold products by selectively transferring reviews from other similar yet warm products. Our work in this paper focuses on this specific problem and generates reviews for cold products through review selection. Similar to existing work in the literature, our work assumes a relationship between product attribute-values and the reviews that products receive. However, unlike the literature, our method (1) is not restricted to the exact surface form of a product attribute name; and, (2) can distinguish between the same attribute expressed in different forms. We achieve these two important characteristics by proposing methods to learn neural product representations that capture the semantics of product attribute-values as they relate to user reviews. More specifically, our work offers (*i*) an approach to learn neural representations of product attribute-values within a shared embedding space as product reviews; (*ii*) a weighted composition strategy to develop product representations from the representation of its attributes; and, (*iii*) a review selection method that selects relevant reviews for the composed product representation within the neural embedding space. We show through our extensive experiments on five datasets consisting of products from CNET.com and movies from rottentomatoes.com that our method is able to show stronger performance compared to several baselines on ROUGE-2 metrics.

1. Introduction

Online shopping is becoming an indispensable part of the modern lifestyle. Access to a variety of competing products from a single point of entry makes online shopping convenient and desirable. In addition to convenience, online shopping platforms enable their users to share their experiences by writing product reviews. This facilitates product purchases by allowing users to make informed decisions. Several studies have reported that product reviews have a direct impact on product conversion rates, i.e., the likelihood of purchase after viewing the product [1,2]. For instance, Askalidis et al. [1] found that when product reviews were available for a product, the likelihood of product purchase increased by 270%. This is even more significant than targeted product advertising. As another example, a recent study showed that 97% of customers make their purchasing decisions on the basis of product reviews such that they read between one to ten reviews before making a purchase [3].

Given the importance of product reviews on product conversion rates, the research community has extensively explored ways in which product reviews can be analyzed and understood. For example, researchers have already worked on measuring product review sentiments [4,5], identifying various aspects of product reviews [6,7], performing review summarization [8] as well as using reviews when making product recommendations [9], just to name a few. However, the challenge with product ratings and reviews pertains to their availability over a range of products. Studies have shown that the availability of ratings and reviews for products follow a long-tail distribution [10]. In other words, there is only a very small fraction of products that has received user ratings or reviews while the vast majority of the other products are not rated or reviewed. The products with an acceptable number of ratings or reviews are often known as *warm or popular products* while the rest of the products are referred to as *cold products*.

* Corresponding author.
E-mail address: bagheri@ryerson.ca (E. Bagheri).

https://doi.org/10.1016/j.knosys.2021.107282
0950-7051/

In order to address this limitation, one of the popular strategies for manufacturers or online retailers is to solicit reviews from professional product reviewers. These professional reviewers often receive product samples ahead of time and publish their independent opinion online, which can help cold products receive more attention. However, in light of the fact that over 80% of products present in online retail websites such as Amazon are *cold* [11], it is practically infeasible to solicit reviews for all such products. As such, an alternative strategy is to generate potential product reviews for a given cold product by inferring its relation to similar warm products. While not as reliable, the process of generating potential product reviews can alleviate the cold product problem.

The basis for existing work that explore ways to generate product reviews is based on the hypothesis that the more two products share similar attribute-values with each other, the more likely it would be for the users to develop a similar perception of these products. These attributes could include product brands, manufacturers, specifications, and date of release, among others. Based on this hypothesis, the work by Park et al. [12] is among the strongest work in this area that employs product specifications to find similarity between warm and cold products and selectively retrieves review sentences from the warm product to be assigned to the cold product. Their model is based on a translation-based information retrieval model, which assumes that the specifications of the cold product are an input query and the search space includes the reviews of similar products. While effective for products with popular and well-known attributes, such a model fails to selectively retrieve appropriate reviews when there is a vocabulary mismatch between product attributes and the words used to refer to them by the users.

Let us further elaborate on this through an example MP3 player product shown in Table 1. In the context of this example, work similar to that of Park et al. consider each of the product attributes as query words in order to identify appropriate reviews. However, when looking at the reviews that are available for the example MP3 player, one can see that many of the product attributes are not explicitly mentioned in the review while being implicitly referenced. For instance, when referencing battery quality, a reviewer can write *'the player runs for days on end without charging'* without mentioning the battery explicitly. Such sentences will not be retrieved and sampled by existing work. On the other hand, there are also other cases when the product attributes do appear in the review but not with the same semantics. For instance, consider a review such as *'I also use it with the Creative Hard Case, which adds thickness, but it's still very pocketable, and the* ***integrated*** *kickstand is a plus'*. With existing work in the literature, the word 'integrated' in the review has the possibility of being matched with the same word in the specification of the MP3 player's flash memory.

As such models that rely specifically on product specifications suffer from (1) the inability to retrieve suitable review sentences that have expressed important relevant information but in different terminology, and (2) the retrieval of semantically unrelated review sentences that have been expressed using homonymous words. The objective of our work in this paper is to address these two main limitations of existing work in the literature. These limitations are primarily due to the focus of existing work on the *keyword-based representation* of product specifications and reviews. As such, we propose a method that would learn product representations that go beyond keywords and captures the semantic relations between product attribute-values and reviews through *soft matching*. More specifically, we propose a method that embeds products and their reviews within a shared multidimensional space where each product will have a dense vector representation and will be placed in the context of the reviews that are most similar to it. In order to learn product and review embeddings, we benefit from the strong literature on neural embeddings in deep learning [13].

1.1. Research objectives and contributions

The objective of our work in this paper is to selectively sample reviews from warm products such that they can be used to generate a potential review for a given cold product. To this end, we address the limitations of existing work that treat products and their specifications through a keyword-based approach by capturing the relation between product attribute-values and product reviews by learning neural embedding-based representations for them such that their relationships are maintained in vector space. In order to do so, we propose to represent each product attribute-value by the reviews of all the products that have such an attribute-value in their specification. The hypothesis of our work is that while such context would inevitably consist of review sentences that pertain to other attribute-values of the products, but given these review sentences are selected from all the products whose main point of commonality is the attribute-value under consideration; therefore, the context will have a higher chance of capturing information about that specific attribute-value compared to other attribute-values. We benefit from this developed context to learn embedding-based representations for reviews and product attribute-values. Now given a cold product, we employ the embedding-based representation of its attribute-values to infer a representation for the product, which would then be used to identify the most related reviews already embedded in the vector space.

More specifically, the main contributions of our work can be enumerated as follows:

1. We propose a model for generating reviews for cold products by selectively sampling reviews from related products without explicitly considering keyword representation of product attributes;
2. We derive product attribute-values and review representations using neural embeddings such that those attribute-value representations that are perceived similarly by the users in their reviews are embedded closer to each other as well as closer to their related reviews within the embedding space;
3. We propose a method to infer the product representation of a cold product based on the embedding representation of its attribute-values, which can then be used for identifying the most relevant and appropriate reviews to be selected;
4. We perform extensive experiments on five real-world datasets collected from the CNET.com and rottentomatoes.com websites and compare our work with the state-of-the-art approaches and demonstrate that our proposed approach is able to outperform existing approaches in terms of ROUGE-2 metrics.

While there is already work, as we will cover in the next section, that perform the task of review selection, our work in this paper distinguishes itself by offering *several innovative aspects*. The first novelty of our work is that it associates product attribute-values with product reviews and hence learns the impact of each attribute value on the overall view of the users for the product. The second distinguishing novelty of this work is that our proposed approach learns how to integrate, in a weighted form, the impact of the set of attribute-values of each product when selecting reviews. In other words, it learns the importance of each attribute-value in the context of other attribute-values. Finally, the proposed approach offers an approach to learn neural representations for attribute-values and product reviews in the same space, which facilitates seamless retrieval of reviews for any given product.

Table 1
A sample MP3 player specification.

Attribute	Manufacturer	Product type	Flash memory	Diagonal size	Battery	Supported digital audio standards
Value	Apple	Digital player	Integrated 64 GB	3.5 in	Lithium Ion rechargeable player battery-integrated	AAC, Audible, AIFF, MP3, WAV

2. Related works

In this paper, we aim to address the problem of generating reviews for cold products. In this section, we will focus on reviewing related works that have worked on different aspects of cold products. There are two main aspects that have been studied by the related work of addressing the cold product problem: rating prediction and review prediction. The overall strategy for addressing the cold start problem has been to consider as much auxiliary information as possible [14–16], such as product purchase or view interactions, product specifications, or knowledge graph data, to make up for the lack of ratings or reviews. In more recent work, some authors have considered using deep learning based techniques to encode product interaction information along with product side information to build product recommendation systems that can be used for cold products. For instance, in our earlier work [9], we have proposed a method, which utilizes unstructured side information such as reviews and product descriptions, as well as transactional information in order to build unique representations for users and products. Wei et al. [17] have proposed a framework to predict product ratings, especially for cold products. They first use a Stacked Denoising AutoEncoder (SDAE) to learn product representations from product specifications, which is then used to measure the similarity between cold and warm products. The obtained representations are then incorporated into a time-aware collaborative filtering framework, namely timeSVD++ [18], in order to make rating predictions.

Given the fact that it is hard to identify ratings or reviews for cold products, many researchers have opted to exploit product specifications of products to handle the cold product problem; however, Zhu et al. in [19] argue that the employment of product specifications for cold products is not the most efficient strategy. To remedy the problem, they propose a recommendation framework, where product attributes are employed in the context of active learning methods in recommender systems. They first pretrain a rating prediction model based on users' historical ratings and product attributes to learn user preferences over product attributes, which represents the 'a priori' perception of users about product attributes. For any new cold product and for a specific user, a rating is predicted based on a priori perception of that user about the attributes of the cold product.

Other authors have considered using a broader range of auxiliary information when handling cold products. In such cases, the use of graph-based representation of products and their auxiliary information has been more common [20–23]. This is particularly because graphs are able to represent the relations between different entities and their attributes and are robust against sparsity and cold start cases [24]. For instance, in the context of the point of interest (POI) recommendation task, Xie et al. [22] have proposed a graph embedding-based approach, which exploits various auxiliary information such as temporal and geographical information. They exploit four bipartite graphs to encode the relationships between the different auxiliary information. Inspired by the LINE method [25], they embed the four heterogeneous information graphs jointly into a shared low-dimension space and represent every user, time slot, geographical region, and POI into a unique vector. Given a user, a time slot, and the current location, the POI with the least distance to them is selected to be recommended to the user. Similarly, in order to utilize various types of auxiliary information, Shi et al. [20] propose to model different product information in a Heterogeneous Information Network (HIN). They propose a random walk strategy over the predefined meta-paths in HIN to obtain meaningful node sequences. Each meta-path provides a representation for nodes by adopting a network embedding method. The resulting representations by each meta-path are further fused to generate a unique representation for users and products. Finally, those representations are incorporated into a matrix factorization model to make recommendations.

The authors in [26] have also considered the idea of meta-paths and proposed to incorporate the mutual impact of meta-paths and user–product interactions by creating a three-way interaction structure in the form of user, meta-path context, and product. To construct a meta-path based context, they propose a priority sampling strategy to select high-quality path instances. Then a deep neural network with a co-attention mechanism is proposed, which leverages the meta-path based context and learns representations for users, products and the meta-path context. The learned representations of user, meta-path context, and product triples are employed to predict product ratings.

Rating prediction task has also been investigated in the context of the product opinion mining, where the sentiments of reviews (either in general or in different aspects of products) are extracted. Techniques for opinion mining are most effective when a reasonable number of products are available and would perform poorly when the number of reviews is low. To address this problem, Moghadam et al. [11] have proposed the Factorized LDA model, which is a probabilistic graphical model based on LDA. It assumes that, in addition to products, users also can be modeled by a set of latent factors. These latent factors are learned using the reviews of all the products that exist in a certain product category, e.g., smartphones. This would include both cold and warm products. For cold products, a priori distributions for each category as well as the rating distribution of the target user or the a prior rating distribution of all users (if the user is also cold) is used to predict ratings. Yang et al. [27] later extended FLDA to take the hierarchy of the product category into account. This approach, known as CAT-LDA, models products in both generic and specialized categories.

While the majority of related studies have been performed for the rating prediction task, there are only a limited number of papers, which address the cold product problem in the field of review prediction. The work in [12] is among the only ones that explicitly focuses on the task of review prediction. This work presents probabilistic retrieval approach for selecting relevant sentences from warm yet similar products for cold products. The authors model the problem of review generation as a generative model that estimates product specifications given a set of product reviews. Given a cold product, this method scores every candidate review sentence as long as the sentence satisfies some predefined criteria, the most important of which is the keyword-based occurrence of the product attribute keywords in the review sentence. The generative model proposed in this work is based on a translation model, which causes it to be sensitive to similar keywords that appear in product specifications and product review sentences. This sensitivity to keywords can impact the retrieval of semantically-similar yet syntactical dissimilar review

Table 2
A comparison between related work that address the cold product problem.

Type	Method name	Citation	Based on textual information	Based on product attributes	Based on temporal information	Based on network-based information	Probabilistic retrieval/ language model-based	Probabilistic retrieval/ translation model-based	Neural embedding-based representation	Based on text summarization techniques	Graph-based	Topic modeling-based
Review prediction	RevSpecGen	[12]		✓				✓				
	Translation	[12]		✓				✓				
	QL	[28]		✓			✓					
	MEAD	[29]								✓		
	MEAD-SIM	[29]		✓						✓		
	TR-SIM	[30,31]		✓						✓	✓	
Rating prediction	FLDA	[11]										✓
	CAT-LDA	[27]				✓						✓
	EBR	[9]	✓		✓				✓			
	Cold	[32]			✓	✓			✓			
	FMFC	[19]		✓					✓			
	IRCD	[17]		✓	✓				✓			
	MCRec	[26]		✓					✓		✓	
	HERec	[20]		✓					✓		✓	
	GE	[22]	✓		✓	✓			✓		✓	

sentences as discussed earlier in the Introduction section of this paper. We have summarized comparison between the related work in Table 2. While there is only the work by Park et al. that explicitly addresses the problem of review prediction, the authors argue that text summarization techniques could be used to predict reviews. In their paper, they adopt such techniques as baselines. We adopt a similar strategy in this paper as well where we adopt such techniques to produce relevant reviews for products for the sake of benchmarking our proposed approach. More details of these methods are provided in Section 5.3 as they are our comparative baselines.

Our proposed method in this paper moves beyond the merely shared keywords between product reviews and product specifications when capturing the similarity between products and reviews. We believe that the representation of each product attribute should not be limited to its syntactic keyword representation but should rather be represented based on how the attribute relates to relevant reviews and other product attributes. To this end, we propose to learn neural-based representations for product attribute-values that are embedded in the same space as reviews. This allows us to use the neural representation of product attribute-values and product reviews to systematically select appropriate reviews based on a cold product's attributes.

3. Approach overview

The objective of our work is to generate/predict reviews for cold products by selectively retrieving reviews from warm products. Our work relies on findings from earlier work, which state that the similarity of structured product specifications (attributes) is a strong indicator of product similarity [9,12]. On this basis, our work relies on product specifications to identify and retrieve relevant reviews. In essence, this is the same basis for which earlier work such as [12,33] operate. However, as mentioned earlier, the work in the literature is limited by only focusing on keyword representation of product specifications. In our work, we propose an approach for learning embedding representations for product attribute-values and product reviews to address this limitation.

Our work is based on one main hypothesis, which is that reviews of a product are generated by the users according to the performance of the product with respect to the various attributes of the product. For instance, reviews that complain about a product being heavy are targeting the weight attribute or reviews that say positive things about image quality are referring to the camera on the product. Based on this hypothesis, our method will selectively retrieve reviews for a cold product based on the attributes that it has. In essence, it will retrieve reviews from other similar yet warm products that have similar attribute-values. For instance, for a cold smartphone with an 8 MP camera and 8 Gb of memory, it will find other warm camera products with similar specifications and will select reviews from such products.

In order to achieve this, we first need to make product attribute-values comparable with each other. To do so, we learn neural embeddings for product attribute-values. We associate each attribute-value with the set of all reviews that are written for products with that attribute-value. This will generate a document for each attribute-value. We then use document embedding techniques over the collection of all attribute-value documents as well as reviews to learn representations for attribute-values and reviews.

Now, given a cold product, we identify the attribute-values that the cold product has and use the corresponding already learnt representations for these attribute-values to build an embedding representation for the product. This embedding representation can then be employed to retrieve the set of reviews that are closest to it in the embedding space.

The overview of our proposed approach is shown in Fig. 1. In the first step and in order to be able to learn embedding representations for product attribute-values and product reviews, we first construct contexts for attribute-values by associating them with product reviews. Given it is not clear what portions of each review are associated with a certain product attribute-value, it would not be possible to build attribute-value contexts by only considering reviews on those attribute-values. As such, we opt to build context for each product attribute-value by associating that attribute-value with all of the reviews that are written for products that have this specific product attribute-value. For instance, the context for the product attribute-value 'Flash Memory = Integrated 64GB' will be the reviews for all those products that have this product attribute and this specific attribute value. This approach for building context can introduce noise because reviews for each product contain information about various attributes of a product. However, our hypothesis is that when reviews for

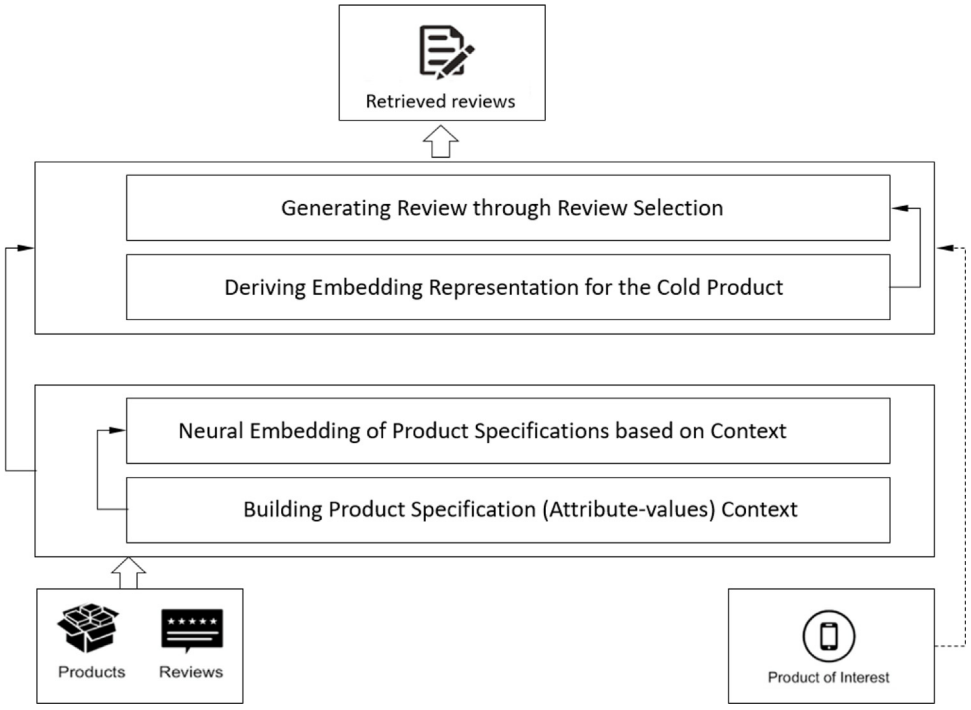

Fig. 1. The overview of our proposed approach.

all products whose only commonality is the specific attribute-value of interest are aggregated, the most frequent aspects of the aggregated text would be about the attribute-value of interest.

Once the context for each attribute-value is built, we will use this representation to learn an embedding based representation. Our work is inspired by methods for the neural embedding of documents. As such product attribute-values and reviews will be embedded in the same embedding space and will be comparable to each other. Considering the fact that we can only assume that the only available information for a cold product is its specification, we can use the embedding-based representation of the attribute-values of each cold product to derive a representation for the product. The derived product representation will be in the same embedding space as attribute-values and reviews and therefore would be rather straightforward to select reviews for the derived product representation according to similarity within the embedding space.

In the following, we will introduce our approach in more detail.

4. Proposed approach for review selection

The main novelty of our work is to learn embedding representation for products and reviews within the same embedding space so that reviews can be effectively sampled based on their vicinity to the product representation. We view the representation of a product as the collective representation of its individual attribute-values. Therefore, we focus on learning embeddings for product attribute-values and then using these to select appropriate reviews. In our work, the process for learning attribute-value and review representations is an *offline* process that needs to be completed only once for a product corpus with its associated reviews. The offline process will build the embedding representations for product attributes-values and reviews. These representations are then used in the *online* process for deriving product representations and for selecting relevant reviews.

We will first describe how product attribute-values are embedded (offline phase) and then subsequently explain our strategy for choosing appropriate reviews (online phase) for cold products. In order to better describe the methods proposed in this paper, we offer a simplified running example with a limited number of reviews and attributes as follows: let us consider a sample product in the camera category namely 'Canon PowerShot SX10 IS'. For this running example, we focus on four attribute-values of this product such as 'Manufacturer:Canon', 'Type:Digital Camera-Compact', 'Resolution:10 Megapixels', and 'Video format:H.264'. We also consider some sample reviews for this product as shown in Table 3, which have been assigned with IDs which are referred to in the following subsections.

4.1. Offline phase: Product specification and review representations

Researchers have already explored various ways of developing product representations, primarily based on word co-occurrence and word frequency using methods such as TF-IDF or BM25 [34]. However, the major assumption of such representation methods is that product reviews are abundantly available for products and hence product representations can be learnt based on product reviews [35–37]. Such methods have shown good performance on different tasks; however, they cannot be applied to cold products. Alternatively, other works [38–40] have explored the use of topic modeling techniques to learn product aspects from across the product review corpus. The limitation of such work is that it becomes difficult to identify the correspondence between specific product attributes and the aspects learned using the topic model. In fact, there may not be a one-on-one match between aspects and product attributes in practice. For instance, several attributes of a product might be considered to form one aspect, which might be suitable for some products (e.g., display and camera on a cheap Smartphone) and not for others (e.g., display and camera on a Professional HD Video Camera).

As such in our work, we are focused on learning individual product attribute-value representations, which could be applied to different products. Our intention is for the product attribute-value representations to be insensitive to the exact use of attribute-value keywords but rather be able to capture the

Table 3
A sample review set for the running example.

ID	The review
R0	This camera provides an attractive option for megazoom shooters.
R1	Shock proof, water proof, big range of built-in shooting settings.
R2	It has a unique tap control.
R3	This camera produces soft photos.
R4	No HD movie capture or raw support.
R5	Horrible picture quality!
R6	This camera, for the price, is the camera that ends all cameras.
R7	Speedy performance with solid battery life.
R8	Articulating LCD; comfortable shooting design; can zoom during movie capture.

semantics of the attribute-values and their relation with product reviews. In the context of our running example, we intend to have a representation space that places the attribute-value 'Resolution:10 Megapixels' close to the review R0:'This camera provides an attractive option for megazoom shooters.' and far from R5:'Horrible picture quality!'.

For this purpose, we adopt a neural embedding strategy, akin to the Continuous Bag of Words (CBOW) model [41], to learn product attribute-value representations specifically because neural embeddings have shown to preserve interesting *geometric properties* as well as *compositionality* [42–44] that can be used for measuring the semantic association between embedded objects. This characteristic of neural embeddings will allow us to (1) semantically reason about the relation between product attribute-values and product reviews, and (2) compose product representations from the individual product attribute-value embeddings.

4.1.1. Building context for product attribute-values

In order to be able to learn product attribute-value embeddings, we will first need to build context for the attribute-values based on which the embeddings will be learnt. Our objective is to learn comparable product attribute-value and product review representations. To this end, we will embed both the reviews as well as attribute-values in the same embedding space. We build context for product attribute-values by associating each attribute-value with the corresponding reviews of all the products that consist of this attribute-value. For instance, in our running example, we aggregate reviews of each product with the attribute-value 'Resolution:10 Megapixels' to create the context for the attribute-value 'Resolution:10 Megapixels'.

Formally stated, given an attribute-value pair (a, v) for attribute a and value v, the reviews of all products whose specification consists of (a, v) will be aggregated:

$$Context_{(a,v)} = \bigcup_{(a,v)\in Spec(P_i)} review(P_i) \tag{1}$$

where P_i is the ith product in the category, and $Spec(P_i)$ is the set of P_i's attribute-value pairs.

The context for (a, v), denoted as $Context_{(a,v)}$, will consist of reviews that are not necessarily exclusively related to (a, v) because all reviews for products with (a, v) are aggregated. However, we hypothesize that while there will be noise from reviews that are related to other attribute-values, but the way we aggregate the reviews increases the likelihood of those reviews that are about (a, v) to be repeated more frequently and hence become the dominating aspect of the representation for (a, v). For instance when (a, v) is 'Resolution:10 Megapixel', we collect all reviews of camera products with the resolution '10 Megapixel' and aggregate them to form the context for this (a, v).

4.1.2. Joint embedding of attribute-values and reviews

The model that we adopt for learning the embeddings is inspired by [45] and assumes that the attribute-value along with a set of context words can be used to predict the next word that will appear in the *context*. Similar to the CBOW model [41], we learn vector representations for every attribute-value as well as every word in its context such that the average of the attribute-value vector and the word vectors can predict the next word in the context. The architecture of the neural network used for learning attribute-value embeddings is presented in Fig. 2.

Formally speaking, let v_t be the unique vector for word t, and v_a denote the vector for the associated context of attribute-value a. Context words in the example of Fig. 2 include Excellent, Sound and Beautiful. The probability of $t + 3$ (Screen in Fig. 2) conditioned over the context is characterized by a softmax function as follows:

$$P(t+3|a) = \frac{\exp(v'^{\top}_{t+3}.v_a)}{\sum_{w=1}^{W}\exp(v'^{\top}_{w}.v_a)} \tag{2}$$

where v'_{t+3} is the output vector representation learnt for $t+3$ and W is the size of the vocabulary in the reviews. Now, we learn the network parameters Θ by maximizing the conditional probability over the whole review collection. Therefore, we have

$$\begin{aligned}\Theta^* &= argmax \prod_{w_t\in T}^{|T|}\prod_{w_c\in C(a)} p(w_t|w_c)\\ &= argmax \sum_{w_t\in T}^{|T|}\sum_{w_c\in C(a)} \log p(w_t|w_c)\end{aligned} \tag{3}$$

where Θ^* is the optimized network parameters, $C(a)$ is the set of context words of a, and T is a randomly sampled set of training words from the set of all words in W drawn for the purpose of negative sampling [42].

Now in order to learn joint representations for both reviews and attribute-values, the corpus used for training the above neural network needs to consist of two sets: (1) a set representing attribute-value contexts; and (2) the set of all product reviews independent from the attribute-values. In order to allow the network to consider both sets, reviews and attribute-value contexts are specified by unique identifiers, representing review and attribute-value IDs corresponding to node a in Fig. 2. Based on this corpus, not only would the attribute-values have corresponding embedding representations, but the reviews will also have embedding representations in the same embedding space as attribute-values. An illustration of such an embedding for our running example is shown in Fig. 3. As mentioned earlier, our goal is to make representations that maintain the semantic relations between attribute-values and reviews. So, based on Table 3, R7 and R8, which discuss Canon-related features such as 'speedy performance' and 'comfortable shooting design' are expected to be close to the representation of the 'Manufacturer:Canon' attribute-value. Also R0 ('provides an attractive option for megazoom shooters') should be close to the 'resolution:10 Megapixels' attribute-value and R3 ('produces soft photos') would be close to the 'resolution:4 Megapixels' attribute-value.

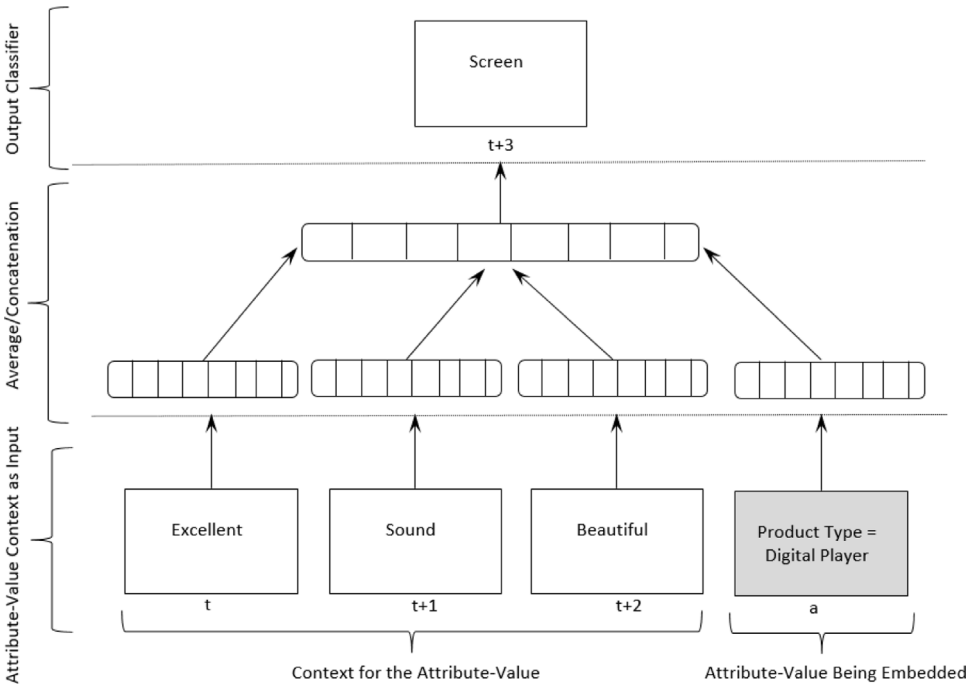

Fig. 2. The neural architecture used for learning attribute-value embeddings.

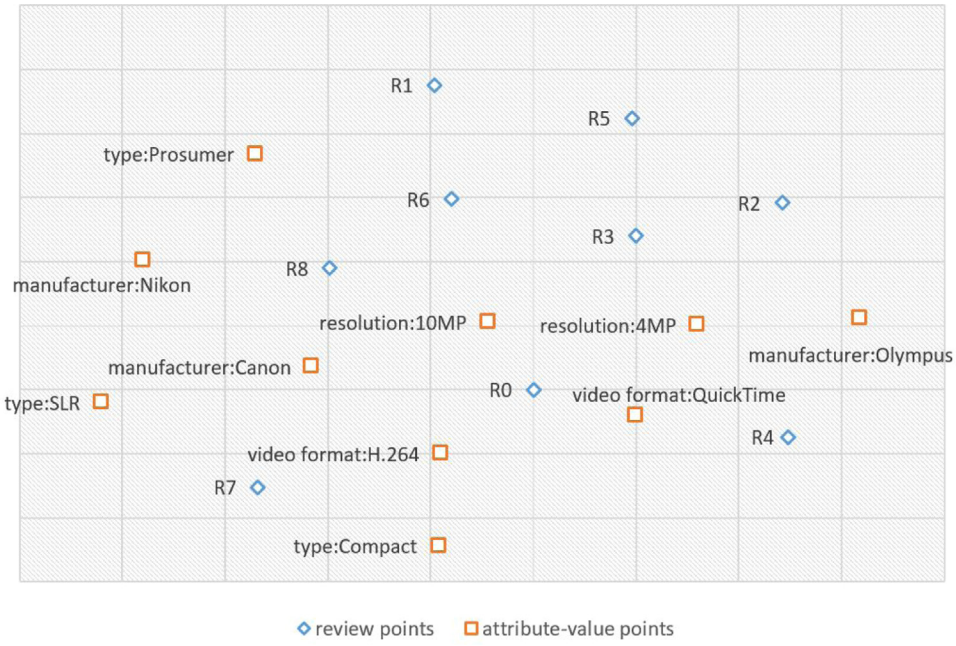

Fig. 3. Joint representing the sample reviews and attribute-values. Review points correspond to reviews in Table 3.

4.2. Online phase: Product representation and review selection

Existing works in the literature on neural embeddings have already considered ways in which multiple embeddings can be integrated. These include concatenation where the embeddings are stringed together or averaging where the element-wise averaging is used to build the new vector. In the context of our work, we use the product attribute-value embeddings to derive a product representation such that the embeddings for the attribute-values of a product are taken into account. The composition of attribute-value embeddings to form a product vector has some specific characteristics that need to be taken into account: (1) different products have a different set of attribute-values and the number of attributes that each product can have might also differ. As such the use of the concatenation approach would not be appropriate as it will generate product representations that have differing lengths. (2) not all attributes of a product have the same importance for the users and hence computing the average of the attribute-value embeddings that assumes that all attribute-values have the same importance is not suitable in this context.

To address these two considerations, we propose to perform a weighted element-wise averaging strategy formalized as follows:

$$v_{P_i} = \frac{\sum_{k\in\{1,\ldots,|A|\}} w_k . v_{s_{i,k}}}{\sum_{k\in\{1,\ldots,|A|\}} w_k} \quad (4)$$

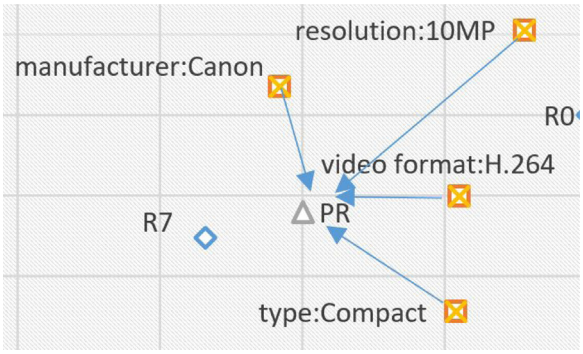

Fig. 4. Product representation (PR) based on attribute-values representations.

where P_i is the cold product of interest, A is the set of attributes, $v_{s_{i,k}}$ indicates the embedding representation for P_i's kth attribute-value pair and w_k shows the weight of the kth attribute. We will discuss later that the weights of attributes can be learnt based on a cross-validation strategy. Fig. 4 shows the obtained representations for the example product 'Canon PowerShot SX10 IS' based on the attribute-values representations, i.e., 'Manufacturer:Canon', 'Type:Digital Camera-Compact', 'Resolution:10 Megapixels' and 'Video format:H.264'.

The advantage of this product representation is that it (1) benefits from the compositionality of attribute-value embeddings, (2) overcomes the limitation of keyword-based models that require exact matches between product specifications and product reviews and (3) captures attribute importances when deriving product representations, (4) allows products with different attribute-value sets to become comparable, and finally (5) places products within the same embedding space as the product reviews. We specifically benefit from this last characteristic of the product embeddings to select appropriate reviews for a cold product.

Given the fact that products are represented in the same embedding space as reviews, we will consider the vicinity of a review to a product as a measure of its *relevance* for serving as a review for the product. In our work, the relevance of a review to a product is measured based on the cosine similarity between their embedding representations. We rank order reviews based on their relevance and generate the final review for a cold product based on the top-k highly relevant reviews.

Table 4
Attributes adopted from CNET.com and rottentomatoes.com for digital camera, MP3 player and movie.

Digital camera	MP3 player	Movie
Manufacturer	Manufacturer	Genre
Product type	Product type	Director
Resolution	Digital storage	Writer
Digital video format	Flash memory installed	Year
Image stabilizer	Built-in display – Diagonal size	
Lens system – Type	Battery/Power – Battery	

We can see in Fig. 5, for the sample product in the running example that reviews R0, R7 and R8 are suitable as the review candidates because they are the most related reviews to the product of interest. So, we would have the generated review for the cold product as 'Speedy performance with solid battery life. Articulating LCD; comfortable shooting design; can zoom during movie capture. This camera provides an attractive option for megazoom shooters'. We note that this is not the actual review generated for this product but is rather based on the simplified running example. Section 6 will provide a discussion on the real reviews generated by our model and qualitatively compare it with the other baselines.

5. Experimental setup

In this section, we present the datasets used to perform our experiments. We also formally introduce the metrics used to measure the performance of our approach against other baseline techniques. We will introduce the baseline techniques employed for comparing our work.

5.1. Datasets

Our proposed approach relies on the structured specification of products. One of the better domains that our approach can

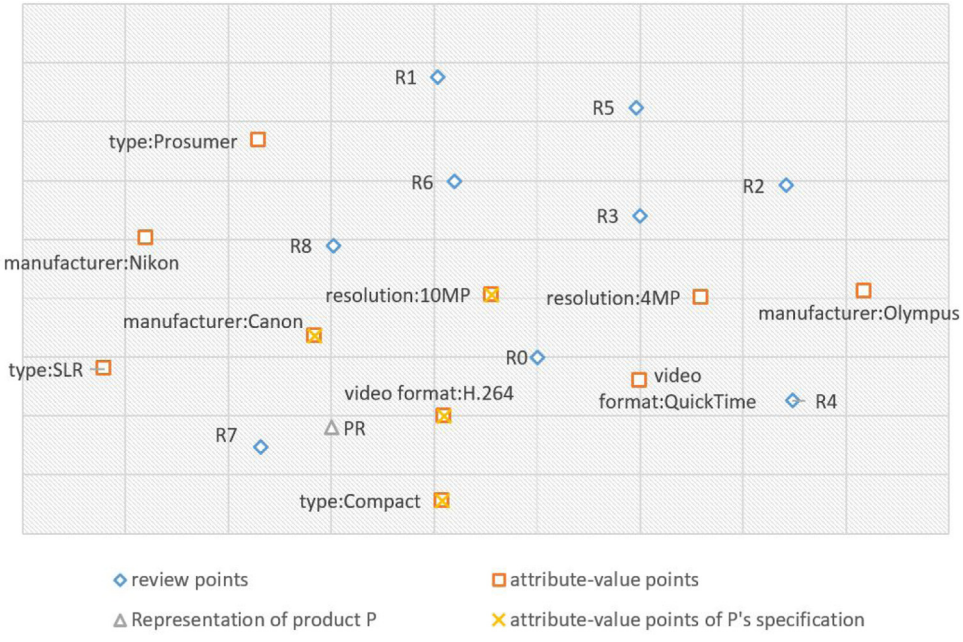

Fig. 5. R0, R7, and R8 are the most related reviews to the product P (point PR).

Table 5
The statistics of the five datasets.

Dataset name	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5
Description	MP3 players in general (warm)	MP3 players products with less than 10 reviews (cold)	Digital cameras in general (warm)	Digital cameras products with less than 10 reviews (cold)	Movies, Documentary based genres, critics reviews
Number of products	605	418	1,153	855	385
Number of users	5,735	1,060	7,506	2,394	1,326
Number of reviews	6,775	1,157	8,856	2,503	6,467

be adopted is e-commerce. In comparison with a domain like movies, products in e-commerce are capable of being described in detail by their specification. As such, any e-commerce platform that contains customer reviews and the structured specification of products can be used for our task. Inspired by the work by Park et al. [12], we adopt the CNET.com website in our experiments. CNET.com contains several product categories, descriptive texts and specifications for its commercial products, as well as customer reviews on the products. As suggested by Park et al. [12], we obtained reviews available in two product categories, namely Digital Cameras and MP3 Players. The data for these categories were crawled from CNET on February 22, 2012. We also selected a dataset from the rottentomatoes.com website, which is a movie review aggregator database that also contains structured specifications of movies. Dataset 5 contains critics' reviews and other relevant content from the top rental playing movies that are available on Netflix for the interval of January 1, 2000, to May 30, 2016. The movies are selected from the documentary-based genre. For each dataset, we adopted the top most frequently available attributes for products in each dataset as shown in Table 4.

As mentioned earlier, our proposed approach is based on neural embedding of product attribute-values and hence can be sensitive to the volume of available review data. To investigate the performance of our work under various review volumes, we adopted two datasets in addition to the datasets suggested by Park et al. According to the definition of the cold products proposed by Moghaddam et al. [11], we consider every product with less than 10 reviews in each category as *cold products*, which form the two additional cold datasets. Since in the movie dataset, most of the movies had received a large number of reviews, unlike the CNET dataset, we did not divide this dataset into two. Table 5 outlines the details of the five datasets.

5.2. Evaluation metrics

To evaluate the performance of our method, the actual reviews of the test products are excluded from the training data. During testing, the generated reviews for test products are compared with the actual ones. This task is similar to the evaluation of summarization techniques, because our proposed review selection approach is similar to the extractive summarization task [46,47]. As such we can adopt commonly used metrics for the summarization task in this context. Metrics for evaluating summarization techniques are based on common n-grams between the actual review and the selected one. ROUGE metrics are widely used in the summarization literature and hence we adopt them for our purpose. To do so, we assume the reviews generated by our proposed approach are the generated summaries and the actual reviews on the product are the reference summaries. ROUGE-2, which is one of the most reliable metrics for evaluating the quality of a summary [48,49], is then used in our evaluation. This metric works based on bi-gram matching and is defined as follows:

$$\begin{aligned} ROUGE2-recall &= \frac{\sum_{bigram\in s} Count_{match}(bigram)}{\sum_{bigram\in r} Count(bigram)} \\ ROUGE2-precision &= \frac{\sum_{bigram\in s} Count_{match}(bigram)}{\sum_{bigram\in s} Count(bigram)} \end{aligned} \tag{5}$$

where s refers to a system-generated review; and r refers to the reference review. $Count_{match}$ denotes the number of common bi-grams between s and r. In cases where several reference summaries are available, ROUGE takes the maximum obtained from the reference summaries. So, we consider the maximum value over r_i, where r_i is the ith reference review.

It is worth mentioning that for the estimation of the performance of summarization systems, a common approach is to evaluate the generated summaries at a maximum length of K words against the reference summaries. Summaries that exceed the size limit will be trimmed down. Since the average length of existing reviews in our datasets is 162, we chose two values for K, i.e., 100 and 200. These values have also been used by the baselines [12] as well. Therefore, we evaluate our work by comparing the top 100 and 200 words of the selected reviews with the actual reviews in terms of the ROUGE-2 metrics.

Table 6
The notation glossary.

t	The candidate sentence
P_z	The product of interest
P_y	The product from which the candidate sentence (t) is derived
P_x	The translation product
S_i	Specification of product i
F	Number of attributes for the products
R	The review set

5.3. Baseline techniques

Generating reviews for new products is an emerging area of research where there are a few works in the literature that consider retrieving relevant sentences for e-commerce products. In this section, we introduce these approaches, which are used as baseline approaches in our experiments. For the sake of readability, we summarize the notation used to introduce the baseline approaches in Table 6. In the following, we provide a brief introduction of these baseline techniques and summarize them in Table 7.

QL: The query likelihood (QL) language model approach [28] is a standard ad-hoc retrieval method that was adopted as the first baseline. The score function in QL is computed as [12]:

$$score(t; R, S_z) = \sum_{k=1}^{F} \prod_{w\in s_{z,k}} p(w|t) \tag{6}$$

where $s_{z,k}$ is the set of words in the kth attribute-value pair in S_z. Here, $p(w|t)$ denotes the probability of the presence of the specification word in the candidate sentence, which follows the unigram language model [12].

Table 7
Descriptions of the baseline approaches.

Method	Citation	Description
QL	Ponte et al. [28]	Query likelihood language model, which scores sentence t based on the similarity between t and the specification of P_z.
MEAD	Radev et al. [29]	Centroid-based summarization technique for multi-document summarization. The method scores review sentences on the basis of their centrality and position in the document.
MEAD-SIM	Radev et al. [29]	Modified version of MEAD, which scores a review sentence on the basis of its centrality and of the similarity between products. The centrality score is computed based on MEAD algorithm.
TR-SIM	Mihalcea et al. [30], Mallick et al. [31]	Modified version of TextRank algorithm, which scores a review sentence on the basis of its centrality and of the similarity between products. The centrality score is computed based on the PageRank algorithm over the sentence graph.
RevSpecGen	Park et al. [12]	Based on a probabilistic generative model wherein each sentence from reviews of a product first generates its specifications. The generated specifications then generate the query specifications.
Translation	Park et al. [12]	Based on a generative probabilistic model in which a selected review sentence will generate the review set of all products, which will be used as the translation of the review. The selected review sentence and each of the generated review sets jointly generate a possible specification for a relevant product related to the selected review.

MEAD: As the second baseline, we use a standard centroid-based multi-document summarization technique, namely MEAD, proposed by Radev et al. [29]. In this method, each review sentence receives a score on the basis of the centroid scores according to the following equation:

$$score(t; R) = w_c C_t + w_o O_t \tag{7}$$

where C_t is the centrality of sentence t and computed by the sum of the centroid scores of the words in t, and O_t is a position score, which assigns higher scores to the review sentences emerging earlier in a document (review). The centroid score of a word is its TF-IDF value in the corpus R, and w_c and w_o are weights for C_t and O_t, respectively. O_t is computed as follows:

$$O_t = \frac{(n - i + 1)}{n.C_{max}} \tag{8}$$

where n is the number of sentences in the document, i is the position of the sentence in the review, and C_{max} is the maximum centroid score in the review.

MEAD-SIM: MEAD selects review sentences on the basis of the centroid score of sentence t, without considering any attributes of the product. Park et al. proposed this modified version of MEAD, by adding the similarity between two products Py and Pz, as follows:

$$score(t; Sy; R, Sz) = SIMp(Py, Pz)C_t \tag{9}$$

where $SIMp(Py, Pz)$ determines the similarity between two products, on the basis of cosine similarity of their structured specification.

RevSpecGen: A common approach in information retrieval is the query likelihood model [50], which assumes that a document generates a query. On the basis of this model, RevSpacGen and the next baseline are the probabilistic approaches proposed in [12], which generate a specification of the product (Sz) from a candidate sentence t via a generative story. The generative story of the RevSpecGen is: a candidate sentence t, which is for product Py, generates Ry^{-t}, denoting the reviews for product y except for t. Then, t and Ry^{-t} are used to jointly generate the specifications for y, indicated as Sy. Then Sy generates the query specification Sz.

$$\begin{aligned} score(t, R_y^{-t}, S_y; R, S_z) &\propto p(t, R_y^{-t}, S_y|S_z) \\ &= \frac{p(S_z|S_y)p(S_y|t, R_y^{-t})p(R_y^{-t}|t)p(t)}{p(S_z)} \\ &\propto p(S_z|S_y)p(S_y|t, R_y^{-t})p(R_y^{-t}|t) \end{aligned} \tag{10}$$

In other words, in this scenario, t is scored based on the similarity between products Pz and Py; the similarity between Py and t and Ry^{-t}, which is computed based on the similarity of their comprising words. The similarity between t and Ry^{-t} is computed based on the TF-IDF cosine similarity of their content.

Translation Model: This is a generative model proposed by [12] where candidate sentence t of a product Py generates each review set of all products, which is employed as the translation of t. The review sentence t and each of the generated review sets, Rx, jointly generate t's specifications Sy; and Sy generates specifications of Rx, Sx, and the query specifications, Sz.

$$\begin{aligned} score(t, S_y; R, S_z) &\propto p(t, S_y|S_z) \\ &= \frac{p(S_z|S_y)\sum_{P_x \in P^{-z}} p(S_x|S_y)p(S_y|t, R_x)p(R_x|t)p(t)}{p(S_z)} \\ &\propto p(S_z|S_y) \sum_{P_x \in P^{-z}} p(S_x|S_y)p(S_y|t, R_x)p(R_x|t) \end{aligned} \tag{11}$$

where P^{-z} refers to the set of all products except for Pz. In other words, in this scenario, t is scored based on the similarity between t and Rx; the similarity between Py, t, and Rx; the similarity between products Pz and Px; and the similarity between products Pz and Py.

TR-SIM: Besides the MEAD summarization approach, we examined another summarization method for selecting review sentences. TextRank is a strong and popular graph-based ranking algorithm for the summarization task proposed by Mihalcea et al. [30] and later used by Mallick et al. [31]. It works on the basis of a sentence graph in which each node represents a sentence and each edge denotes the similarity of the corresponding nodes. In the TextRank approach, when one node links to another one, it is basically casting a vote for that node. The higher the number of votes cast for a node, the higher the importance of that node. This is the basis of the PageRank [51] algorithm as well. So the centrality of the sentences in the TextRank algorithm, R_t, is computed on the basis of the PageRank score of the corresponding nodes in the sentence graph.

While MEAD-SIM selects a review sentence on the basis of the centroid score of the sentence, i.e., C_t and the similarity between two products Py and Pz, TR-SIM scores sentences based on the rank of the sentence R_t and the similarity between two products Py and Pz as follows:

$$score(t; Sy; R, Sz) = SIMp(Py, Pz)R_t \tag{12}$$

where $SIMp(Py, Pz)$ determines the similarity between two products, on the basis of cosine similarity of their structured specification.

Table 8
Time required for training EbRS, MEAD-SIM and TR-SIM.

	Dataset1	Dataset2	Dataset3	Dataset4	Dataset5
Execution time for training EbRS	64.41 s	56.53 s	232.57 s	77.10 s	42.84 s
Average execution time for selecting reviews for each product in EbRS	0.92 s	0.90 s	2.04 s	2.03 s	1.64 s
Execution time for training MEAD-SIM	894.01 s	24.65 s	772.25 s	65.93 s	48.38 s
Average execution time for selecting reviews for each product in MEAD-SIM	392.24 s	14.84 s	166.20 s	24.53 s	8.06 s
Execution time for training TR-SIM	224.84 s	892.07 s	682.36 s	24532.57s	458.16 s
Average execution time for selecting reviews for each product in TR-SIM	392.24 s	14.84 s	166.20 s	24.53 s	8.06 s

6. Evaluation results and findings

In this section, we evaluate the performance of our work in comparison with the baseline methods on the five datasets collected from CNET.com and rottentomatoes.com. We randomly selected 50 products from the dataset and excluded them as well as their reviews from the training set. These products are then used in the test set. For the preprocessing task, i.e., stopword removal, tokenization and stemming, we used the natural language toolkit provided by the NLTK library in Python.

6.1. Training time

One of the important questions in building the product and review representation models is to see how these representations are updated with new data. There are two main types of representations introduced in the proposed approach, review representation and attribute-value representation. Both representations are generated on the basis of the reviews in the systems where new reviews are constantly received. It is also possible for new attribute-values to emerge in product attributes. In order to build representations for new reviews and attribute-values, as well as updating the older representations, we need to retrain the model. Table 8 reports the time required for training the EbRS model on our five datasets.[1] As seen in the table, the required time of EbRS is quite small, due to the highly scalable nature of neural embedding methods. Therefore, it is feasible to re-learn the reviews and attribute-value representations on a periodic basis. Here, we also discuss the training times of the other competitive baselines, i.e., Translation, MEAD-SIM and TR-SIM. Due to the high amount of probabilistic computations in the Translation model, we had to distribute the training over 40 cores. Nevertheless, it took from 2 to 8 days to complete the training process depending on the size of the dataset. Due to the big difference, We did not include the Translation model training time in the table. We also report the training times of the MEAD-SIM and TR-SIM in this table which are quite short in comparison with the Translation model but still, we can see that these baselines are slower than EbRS.

Here, it is worth mentioning that the main factor which affects the training time of EbRS is the size of the training corpus introduced in Section 4.1. The corpus comprises two types of documents, reviews and attribute-value contexts. Referring to Table 5, Dataset 3 has the highest number of reviews and the highest number of products. The higher the number of reviews is, the higher the number of review documents in the training corpus would be. A higher number of products implies a higher number of attribute-values and this results in a higher number of attribute-value contexts. As we see in Table 8, Dataset 3 has the highest training time among other datasets. Similarly, it is understandable that Dataset 2 with the lowest number of reviews and products has the lowest training time. Overall, the running time of the proposed method is quite short and scalable to large datasets.

[1] The reported times are based on 2×2.7 GHz Eight-Core Intel Xeon Processor with 20 MB Cache–E5-2680 with 256 GB of memory running Ubuntu 16.

6.2. Model tuning

One of the aspects of our proposed method which can affect the performance of the review selection process is the vector size of the embeddings. In our experiments, we examined various vectors sizes including 50, 100, 150 and 200. We found that a vector size of 50 shows the best results across all our five datasets; however, the differences are not statistically significant. The second aspect is the weights assigned to attributes for deriving product representation as discussed in Section 4.2. In order to identify attribute weights, we adopt a five fold cross validation strategy. We examine weights within the range of [0,1] for each attribute with an interval of 0.25. The optimal weights for product attributes were learnt based on this strategy for each dataset separately.

6.3. Evaluation results

The evaluation results of the baselines in terms of ROUGE-2 precision, recall, and F1-score metrics are reported in Tables 9–12. In this section, we refer to our proposed approach as Embedding-based Review Selection (EbRS). As mentioned earlier, we tried different variations of our work, each of which with a unique attribute weighting combination. Boldface numbers reported in the tables indicate that the improvement shown by EbRS is statistically significant based on a paired t-test with a confidence interval of 95% against the best performing baseline method.

As shown in Tables 9–13, our proposed approach has outperformed the other baselines in most cases. The results in Table 9 show that EbRS significantly outperformed the other baselines on the MP3 Player dataset, in terms of ROUGE-2 precision and F1-score metrics. However, the results in this table show that the improvement shown by EbRS on Dataset 1 in terms of the recall metric was not significantly significant although the average performance was higher than other baselines. Furthermore, the results in Table 10 show that our approach, in terms of the ROUGE-2 precision metric did not outperform the baselines on the Dataset 2, i.e., cold MP3 Player products. When looking into the specifics of Dataset 2, we find that this dataset contains the lowest number of reviews compared to other datasets. As a result, the corpus derived from this dataset for the purpose of review and attribute-value embedding is rather small. Given our approach relies on a neural embedding that favors large training data, the lower performance on precision can be attributed to the quality of the extracted attribute-value embeddings. Having said that, the proposed method still provided statistically significant improvement over the baselines on recall of both @100 and @200 as well as F-score of @100.

We can see in Table 11 that EbRS significantly outperformed other approaches on Dataset 3 relating to Digital Cameras on

Table 9
The performance of EbRS in comparison with the baselines on Dataset 1 on MP3 players. Boldface values indicate statistical significance.

	Precision		Recall		F-Score	
	@100	@200	@100	@200	@100	@200
EbRS	**0.098**	**0.084**	**0.134**	**0.160**	**0.113**	**0.110**
Translation [12]	0.060	0.054	0.108	0.153	0.077	0.079
QL [28]	0.052	0.038	0.080	0.113	0.063	0.057
MEAD [29]	0.036	0.029	0.081	0.091	0.024	0.043
MEAD-SIM [29]	0.050	0.047	0.090	0.110	0.064	0.065
RevSpecGen [12]	0.063	0.055	0.100	0.138	0.077	0.078
TR-SIM [30,31]	0.055	0.046	0.095	0.115	0.069	0.065

Table 10
The performance of EbRS in comparison with the baselines on Dataset 2 on cold MP3 players. Boldface values indicate statistical significance.

	Precision		Recall		F-Score	
	@100	@200	@100	@200	@100	@200
EbRS	0.072	0.054	**0.132**	**0.145**	**0.094**	0.079
Translation [12]	0.082	0.063	0.100	0.135	0.090	0.085
QL [28]	0.044	0.036	0.071	0.104	0.055	0.053
MEAD [29]	0.040	0.037	0.130	0.160	0.050	0.051
MEAD-SIM [29]	0.050	0.047	0.090	0.100	0.064	0.064
RevSpecGen [12]	0.054	0.047	0.970	0.127	0.069	0.068
TR-SIM [30,31]	0.059	0.036	0.095	0.116	0.072	0.054

Table 11
The performance of EbRS in comparison with the baselines on Dataset 3 on digital cameras. Boldface values indicate statistical significance.

	Precision		Recall		F-Score	
	@100	@200	@100	@200	@100	@200
EbRS	**0.066**	**0.048**	**0.140**	**0.177**	**0.090**	**0.076**
Translation [12]	0.040	0.040	0.091	0.130	0.055	0.061
QL [28]	0.043	0.038	0.077	0.100	0.055	0.055
MEAD [29]	0.058	0.033	0.049	0.076	0.053	0.046
MEAD-SIM [29]	0.052	0.047	0.110	0.150	0.072	0.071
RevSpecGen [12]	0.033	0.028	0.075	0.123	0.046	0.046
TR-SIM [30,31]	0.060	0.046	0.119	0.158	0.079	0.071

Table 12
The performance of EbRS in comparison with the baselines on Dataset 4 on cold digital cameras. Boldface values indicate statistical significance.

	Precision		Recall		F-Score	
	@100	@200	@100	@200	@100	@200
EbRS	**0.076**	**0.052**	**0.147**	**0.182**	**0.100**	**0.080**
Translation [12]	0.049	0.038	0.110	0.141	0.067	0.059
QL [28]	0.029	0.044	0.055	0.099	0.037	0.041
MEAD [29]	0.058	0.033	0.106	0.160	0.053	0.046
MEAD-SIM [29]	0.059	0.050	0.120	0.150	0.079	0.075
RevSpecGen [12]	0.031	0.024	0.084	0.123	0.046	0.040
TR-SIM [30,31]	0.059	0.050	0.120	0.157	0.079	0.075

all metrics. This means that when a larger number of reviews are available, the proposed method would have an acceptable performance in representing attribute-values and products and selecting proper reviews for cold products. This is consistent with our assumption that when a large corpus exists, reviews that are written for a product with a specific attribute-value, prepare a proper context for representing that attribute-value. This was also observed in the results of Dataset 1 for MP3 Player products, which is also a non-cold dataset. In addition, our approach outperformed the baselines on all metrics for Dataset 4. Comparing Dataset 2 with Dataset 4, we can see that the size of Dataset 4 is larger than Dataset 2 (2,503 reviews compared to 1,157 reviews). This again confirms our explanation that our method performs well when a larger number of reviews exist to learn the attribute-value and review embeddings. We can also observe from Table 13 that EbRS performs more effective than other baselines on the movie dataset in terms of all metrics.

Having said that we would like to clarify how the size of the dataset impacts the performance of our proposed approach. The assumption of our work is that there are reviews available for warm products that will be used to learn representations for attribute-values, which can be used to generate reviews for cold products. We find that our model performs well when there is a larger number of reviews for the warm products for learning the attribute-value embedding representations. This is why our proposed approach is able to significantly improve on the baselines in Datasets 1, 3, 4, and 5 but only partially improve them on Dataset 2, which is smaller in size compared to the four other datasets.

It is worth exploring the reasons behind why the results on Dataset 5 are not as strong as the other datasets in general for all of the methods. We find that the main reason arises from the difference between the nature of commercial products and movies. First, the movie domain consists of a wider range of

Table 13
The performance of EbRS in comparison with the baselines on Dataset 5 on movies. Boldface values indicate statistical significance.

	Precision		Recall		F-Score	
	@100	@200	@100	@200	@100	@200
EbRS	**0.013**	**0.010**	**0.100**	**0.147**	**0.023**	**0.019**
Translation [12]	0.010	0.008	0.072	0.112	0.018	0.014
QL [28]	0.004	0.005	0.053	0.060	0.007	0.009
MEAD [29]	0.012	0.007	0.085	0.095	0.019	0.013
MEAD-SIM [29]	0.011	0.008	0.092	0.116	0.020	0.014
RevSpecGen [12]	0.009	0.008	0.065	0.092	0.016	0.015
TR-SIM [30,31]	0.012	0.008	0.091	0.115	0.021	0.015

Table 14
Impact of weights on the performance of the EbRS.

EbRS Performance (ROUGE-2)	Dataset 1		Dataset 2		Dataset 3		Dataset 4		Dataset 5	
	@100	@200	@100	@200	@100	@200	@100	@200	@100	@200
Precision-without weights	0.066	0.063	0.065	0.050	0.034	0.039	0.047	0.041	0.006	0.006
Precision-weighted	0.098	0.084	0.072	0.054	0.066	0.048	0.076	0.052	0.013	0.010
Recall-without weights	0.092	0.127	0.089	0.114	0.072	0.160	0.109	0.158	0.039	0.039
Recall-weighted	0.134	0.160	0.132	0.154	0.140	0.177	0.147	0.182	0.100	0.147
F1-Score-without weights	0.077	0.084	0.075	0.070	0.046	0.063	0.066	0.065	0.010	0.010
F1-Score-weighted	0.113	0.110	0.094	0.079	0.090	0.076	0.100	0.080	0.023	0.019

items compared to the e-commerce domain. In other words, the diversity of movies in a specific genre is greater than the diversity of products in a specific category. Second, while an e-commerce product can be described by its features to a great extent, there are some implicit factors in each movie that affect the users' opinions which are not captured in the structured specifications of the movie [19]. In other words, it is possible to have movies with similar structured specifications towards which users have different opinions. As such transferability of reviews based on structured specifications is possible to a lesser extent in the movie domain compared to the e-commerce domain.

6.4. Impact of weights

In Section 4.2, we introduced the EbRS strategy for driving product representations. We mentioned that we exploit the product attribute-value embeddings to derive a product representation and explained why we used a weighted strategy over the attribute-value embeddings to build the product representation according to Eq. (4).

In this section, we show the impact of considering weights on the performance of EbRS. To do so, we examine the performance of the proposed method in cases where no weighting scheme is used. The results of these experiments and the comparison between the weighted version of EbRS are reported in Table 14. As shown in Table 14, the impact of using weights on improving the performance of EbRS is significant on all five datasets in terms of our various evaluation metrics. This is an indication that not all product attributes contribute in the same way to the users' perception of a product, and hence, a weighted strategy would allow us to capture the importance of each attribute for review generation. This leads to more accurately generated reviews for each product.

6.5. Rating analysis

In this section, we experiment how the reviews from our method compare to the best baseline when used to predict product ratings. To do so, we predict the rating of a product based on the reviews that are generated by our method and the best baseline and compare it with the average user rating of the product of interest. For this purpose, we adopt the state of the art rating prediction architecture based on product reviews that was proposed and successfully used in the Kaggle Toxic Comment Classification competition [52]. This architecture has a combination of LSTM (long short-term memory) layers and GRU (gated recurrent units) layers. In our experiments, reviews are classified into five classes (equivalent to product ratings of 1, 2, 3, 4, 5). The predicted ratings of reviews are compared with the average actual ratings of product given by the users. We estimate the effectiveness of our model against the best baseline in terms of the Root Mean Square Error (RMSE) metric. Formally,

$$RMSE = \sqrt{\frac{\sum_{(p)\in P_{test}} (r_p - \hat{r}_p)^2}{|P_{test}|}} \tag{13}$$

where r_p and $\hat{r}_p$ represents the average of the actual ratings of product p and the predicted rating based on the generated review for that product, respectively; and P_{test} denotes the set of test products. The lower the RMSE values is, the better the quality of rating prediction would be.

The performance of our proposed method compared to the best baseline, i.e., Translation [12] (determined according to the performance of the baselines reported in Section 6.3) is shown in Table 15 based on a five fold cross validation strategy. We observe that our proposed model outperforms the baseline on all datasets, except on Dataset 2. This observation is consistent with the observed results of the ROUGE metrics in Section 6.4. We discussed there that due to the small size of Dataset 2, the neural embedding method has lower performance compared to its performance on the other datasets. However, our model outperforms the baselines on the other four datasets in terms of RMSE metric. This clearly indicates that higher quality reviews lead to a more accurate prediction of the product rating and therefore can have practical application not only to bootstrap products with initial reviews but also for proactively predicting possible product ratings based on the generated reviews.

6.6. Discussions and qualitative analysis

In this section, we show a qualitative comparison between our proposed approach and the baseline methods by investigating sample selected reviews from EbRS, translation, MEAD-SIM and TR-SIM approaches. We choose MEAD-SIM, TR-SIM and translation approaches because they have better results in our experiments compared to the other baselines. We discuss how

Table 15
Comparison of the performance of EbRS and translation models in terms of RMSE metric.

	Dataset 1		Dataset 2		Dataset 3		Dataset 4		Dataset 5	
	@100	@200	@100	@200	@100	@200	@100	@200	@100	@200
EbRS	1.03	1.02	1.05	1.08	0.77	0.77	0.87	0.73	1.26	1.32
Translation [12]	1.22	1.09	0.88	0.93	1.0	1.16	1.16	0.97	1.28	1.35

and why the selected reviews are appropriate and where they differ from the actual reviews. We base our discussions in this section on Table 16, which shows the outputs generated for an MP3 player in Dataset 1, namely 'Apple iPod (30GB, video)', which is a cold product in our experiments.

According to the review written by the CNET editor on this product, there are some negative points for this product as follows: *No extras included, such as a dock, A/V cables, or a power adapter; poor battery life for video;* By looking at the customer reviews, we can see that these features are discussed in the majority of customer reviews as well. When looking at the EbRS selected reviews for this product, we can see that these points are covered in the top selected sentences. Another interesting observation is that the semantics of the retrieved reviews is aligned with the actual review of the editor. For instance, EbRS sentence: *I do, 2 h of battery life with video playback is terrible.* matches with this actual review: *TWO HOUR BATTERY LIFE!!!*. Another example of this matching is seen in this selected review by EbRS: *apple quality, itunes, slim design, good screen specs for it is size* and its resemblance to the actual review: *use of dials & slim design are excellent!*

Another interesting observation about EbRS being successful in capturing the semantics of product reviews can be seen in the second review selected by EbRS, i.e., *poor video playback battery life, no div x support, video out only through dock. 'so far, we don't see any immediate weaknesses with the new apple ipod' - james kim*. The reviewer quotes a sentence from James Kim who had originally introduced the product. The reviewer intends to convey that although James Kim claims that the product has no weaknesses, it has weaknesses such as poor video playback and battery life. In other words, despite the positive words in the final sentence, we can understand that the whole review stands on negative ground. EbRS is able to capture the semantics of this review and appropriately select and retrieve it.

Regarding the output of the translation model, we can observe that several features are mentioned in the selected sentences, such as 'video', 'design small', 'sleek design', 'radio', 'sound quality', 'click wheel', 'flow view', 'battery life', 'camera quality', and 'lens place'. However, in several cases, the mentioned feature is not described correctly in the selected sentences. For example, while the feature 'battery life' is an important feature in the actual reviews, it appears as the sixth sentence retrieved by the translation model ('good battery life, camera doesn't have good quality, and the lens is in a very awkward place.') and consists of a review for this product that does not seem to be the representative of what the users thought about this product in general. Moreover, we can see the second sentence of the translation model tells us about the overall quality of the product which is not consistent with the actual evaluation by the users. Both MEAD-SIM and TR-SIM work based on the centrality of sentences. While MEAD-SIM uses the centroid score and TR-SIM exploits the PageRank score, the outputs of these approaches are very similar. We can see that for the mentioned product, the top first sentences are the same. MEAD-SIM and TR-SIM have similar problems to the translation model since these approaches rely on matching syntactical words of the product specifications. We can see several attributes detected by MEAD-SIM and TR-SIM in the first sentences such as 'processor', 'rear camera', 'display', 'video quality', 'microphone', 'dock connector' and 'speaker'. However, the sentences selected by MEAD-SIM for describing these attributes of the product are inconsistent with the actual reviews. Moreover, some of the sentences include attributes that are not considered to be important from the users' perspective, such as the one focusing on the 'rear camera' attribute that has not actually been mentioned by the users in their reviews for this product. This is despite the fact that MEAD-SIM and TR-SIM have not retrieved reviews for more important features such as 'battery life'.

We find that this shortcoming could be because the products in the top sentences of MEAD-SIM, TR-SIM and translation include products whose quality, such as battery life, and sound quality is not close to that of the product of interest. For example, the translation model retrieved some irrelevant sentences for the mentioned cold product, which were selected from reviews of products that are in the same family but have different ratings from the users. For instance, 'Apple iPod Touch (second generation, 16GB)' with an overall rating of 8.5 and 'Apple iPod Classic (160GB)' with an overall rating of 6.4 are two products selected by the translation model that have completely different reviews compared to the cold product of interest. As another example, TR-SIM selects the same sentences for two products 'Apple iPod Touch (first generation, 8GB)' with an average rating of 8.3 and 'Apple iPod Shuffle (third generation, 4GB)' with an average score of 4.1. Another limitation of the MEAD-SIM, the TR-SIM and EbRS approaches is that they select the exact same review sentences for products with similar specifications. These duplications happen because these approaches are heavily focused on the specifications of the products. As a result, when the similarity of the structured specification is high, the selected reviews for the products tend to become more similar. However, there is an important difference between EbRS and MEAD-SIM/TR-SIM. MEAD-SIM and TR-SIM compute this similarity syntactically while EbRS computes this similarity semantically. In other words, the similarity metric in MEAD-SIM and TR-SIM relies on the common words between words specifications, while similarity in EbRS is defined on the basis of the similarity of the semantic representations of the attribute-values. As a result, MEAD-SIM and TR-SIM are weaker than EbRS in distinguishing two products that are highly similar in their attribute-value words, while different in their reviews. For instance, let us consider the two products and their attributes in Table 17.

Consider the values in the attributes 'Manufacturer', 'Product type', 'Battery', and 'Flash memory installed' in $P1$ and $P2$. Since there are several common words between these values, the similarity computed by the MEAD-SIM approach is high. Consequently, based on the transitive property of the similarity metric, these two products would have many similar products in the training set. As a result, common sentences would be selected by the MEAD-SIM approach for $P1$ and $P2$. However, EbRS takes every distinct attribute-value pair as a unique item that needs to be embedded. As a result, EbRS does not consider the mentioned attributes 'Battery', and 'Flash Memory' as similar attributes, solely because they have several common words in them (64 gb integrated vs 4 gb integrated). Consequently, the selected reviews for these products by EbRS will not be the same, unlike MEAD-SIM. Let us consider the actual reviews for $P1$ and $P2$ written by an expert reviewer called Donald Bell:

P1: *The third generation of Apple's iPod Touch is still the king of the hill when it comes to portable, Wi-Fi-wielding media players. New additions such as Voice Control, graphics enhancements,*

Table 16
Reviews selected by EbRS, Translation, MEAD-SIM and TR-SIM approaches for the 'Apple iPod (30GB, video)' product.

EbRS	Translation [12]	MEAD-SIM [29]	TR-SIM [30,31]
A good player but everyone else did it 6+ months ago. apple quality, itunes, slim design, good screen specs for it is size Poor video playback battery life, no div x support, video out only through dock. "so far, we don't see any immediate weaknesses with the new apple ipod" - james kim 2 h of battery life with video playback is terrible. With itunes now going to have desperate house wives etc. The ipod video cannot be seen as merely meant for music video clips, so there is no excuse for the poor battery life. This player is definitely not for long trips — especially since there is still no removable battery. An fm tuner is still not present. and tv out only through the dock is a downer. However the ultra slim design is impressive and itunes video is sure to make this a hit with the less 'tech savvy.	It does not show all rectangular shaped album artwork! Best chunk of aluminum I ever had the fortune to buy. Reply this 5th revision of the ipod brought video to the mix. design small, sleek design A plethora of features, including radio and video camera excellent sound quality. Click wheel is very responsive and does not freeze retains cover flow view and shake feature from previous generations. Good battery life, camera does not have good quality, and the lens is in a very awkward place. Camera cannot take photos capacity is still stuck at either 8 gb or 16 gb this is by far my favorite nano yet.	Actual owner of ipod touch 4g. The display is brilliant, the speed of the processor amazing, video quality is very much enjoyable and the rear camera is excellent for taking quick fun shots. Internet browsing loads super fast. No in-line microphone on the earbuds. Dock connector does not sit flush with the device. Speaker gets blocked easily alright, so best buy got the 8 gb ipod touch 4g in stock so I drove an hour to go get on. I was not disappointed. I will run through the features. I have come across so far. Body: aside from moving the sleep button to the right and making the back of the device slimmer and more flat, not much has change.	Actual owner of ipod touch 4g. The display is brilliant, the speed of the processor amazing, video quality is very much enjoyable and the rear camera is excellent for taking quick fun shots. Internet browsing loads super fast. No in-line microphone on the earbuds. Dock connector does not sit flush with the device. Speaker gets blocked easily alright, so best buy got the 8 gb ipod touch 4g in stock so I drove an hour to go get on. I was not disappointed. I will run through the features. I have come across so far. Body: aside from moving the sleep button to the right and making the back of the device slimmer and more flat, not much has change.

Table 17
Two sample products considered to be similar by MEAD-SIM and TR-SIM.

	Product name	Manufacturer	Product type	Flash memory	Digital storage	Battery	Overall rating
Product 1 (P1)	Apple iPod touch (third generation, 64 GB)	Apple	Digital player	64 GB integrated	Digital storage	Lithium ion rechargeable player battery integrated	6.9
Product 2 (P2)	Apple iPod shuffle (third generation, 4 GB)	Apple	Digital player	4 GB integrated	None	Lithium polymer rechargeable player battery integrated	4.1

improved accessibility, higher capacity, and a faster processor help to refine an already excellent product. The video cameras found on the iPhone 3GS and iPod Nano remain conspicuously absent. The lack of refinements to its audio quality, hardware design, and video playback leave us feeling just a little uninspired. Though the updates are subtle, the third-generation iPod Touch leaves its competitors in the dust.

P2: *The third-generation Apple iPod Shuffle is as light as a feather and as small as a paper clip, and it includes VoiceOver cues, and improved support for podcasts and audiobooks. You need to operate the Shuffle using a pill-size remote control on your headphones, battery life is not great, features are few, and the design is a bit boring. The third-generation iPod Shuffle is the next best thing to an invisible MP3 player, but the quirky controls and microscopic design make it a limited recommendation.*

These reviews represent the perspective of an expert user. Ideally a review generation method should generate reviews that are similar to them. We also report the reviews from our proposed approach, EbRS:

P1: *Amazing, perfect, everyone loves it. Large touch screen, flawless design (hardware and software), app store, wifi, youtube, movies, perfect function, motion. Scratch magnet back, battery is not the greatest, umm... cannot think of much else. I got this from costco a few weeks ago as they only had the 32gb one in stock, and now I am glad I put in the extra cash for the extra space.*

P2: *pretty good player for a decent price. screen protection, radio, voice recorder, scroll wheel, build. non-customizable interface. My cousin just threw the black fuze over to me after getting an ipod and I honestly do not know why he would do that since this is more capable than an ipod. I really like the build of it, how it has a nice shiny but not reflective face and a brushed metal back, it feels sturdy. I also loved the scroll wheel. I just like the sound of the wheel when it turns for some reason and there is a nice ring of light around it. Sony is a genius sound quality, earbuds, interface, battery life, pricing, pretty thin.*

In contrast, MEAD-SIM generates the exact same review for both products as follows:

P1 and ***P2***: *Actual owner of ipod touch 4g, display is brilliant, speed of the processor amazing, video quality is very much enjoyable, rear camera is excellent, internet browsing loads super fast, no in-line microphone on the earbuds, dock connector does not sit flush with the device, speaker gets blocked easily. Alright, so best buy got the 8gb ipod touch 4g in stock so I drove an hour to go get on. I was not disappointed. I will run through the features I have come across so far. Body: aside from moving the sleep button to the right and making the back of the device slimmer and more flat, not much has change yes, the chrome back is still there it was so pretty for the first 30 s ram: the ram on the ipod touch is only 256 mb, so do not believe the talk of 512 mb, it is simply not true. wireless: fully supports.*

In practice and when comparing MEAD-SIM and EbRS, we observe that MEAD-SIM returns the same sentences for $P1$ and $P2$, while EbRS returns different reviews consistent with the product quality primarily because EbRS is able to embed and capture the semantics of product attributes in contrast to MEAD-SIM which only relies on term matching over product attributes.

7. Concluding remarks

In this paper, we have focused on generating reviews for cold products by selectively sampling relevant reviews from warm

Table 18
The summary of comparison between main review generation methods.

	Speed	Cohesion	Semantic-enabled	Precision	Recall	Multiple features overlay
EbRS	Very high	High	High	Medium	Medium	Medium
Translation	Low	Medium	Medium	Low	Low	high
MEAD-SIM	high	Medium	Medium	Low	Low	High
TR-SIM	high	Medium	Medium	Low	Low	High

products. The innovative aspect of our work is that we learn neural representations for product attribute-values and reviews within the same embedding space. We then rely on the *compositionality* and *geometric* properties of neural embeddings to learn representations for actual products that are composed of individual attribute-values. Given products and reviews are embedded in the same representation space in our work, we select reviews that are most related to the cold product of interest when generating reviews. We have benchmarked our proposed approach against several strong baselines based on products from the CNET.com website. We find that our method is able to show improved performance compared to the baselines in terms of ROUGE-2 metrics, especially when the size of the warm product corpus is large. We have summarized our observations in Table 18.

There are areas where our work could be improved, which we are interested in exploring as a part of our future work:

1. Our proposed method is dependent on the availability of a sizeable amount of reviews for warm products that can be then transferred onto the cold products. In other words, while it does not require any reviews for the cold products, it does expect reviews for warm products. This could be considered to be a limitation for domains which are considered low resourced domains, e.g., languages that do not have much content related to product reviews. We are interested in exploring how reviews can be generated based on sequence models.
2. Similar to most other baselines, our model relies on product attributes to associate products to reviews. This will work very well for domains where product attributes are properly defined and accessible. However, for some other platforms, which consist of mostly unstructured product content, such as craigslist, this would be seen as a limitation. We are interested in exploring whether product images could be used as a complement or replacement for the need to have well-defined product attributes.
3. In our work, we do not consider user–product or user–review associations. Such information could potentially improve the review selection process as issues of user reliability and trustworthiness could be taken into consideration.
4. Finally, like most research work in this area, the motivation for this paper has been to generate reviews for cold products with the hopes of attracting more attention to them within the ecommerce platform. Increasing the likelihood of product views through the availability of reviews increases the chances of the product being purchased by the users. Such a strategy can lead to benefits for sellers. However, this paper does not explore the impact of such review generation process on the buyers. While we can speculate that accurate reviews will be helpful to users for predicting the quality of products, the quantifiable impact is not measured in this work. As future work, We are interested in running an empirical study, which would measure the impact of product review generation on customer satisfaction under controlled settings.

CRediT authorship contribution statement

Fatemeh Pourgholamali: Conceptualization, Software, Validation, Investigation, Formal analysis, Writing – original draft. **Mohsen Kahani:** Investigation, Project administration. **Zeinab Noorian:** Conceptualization, Validation. **Ebrahim Bagheri:** Conceptualization, Investigation, Methodology, Software, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Georgios Askalidis, Edward C. Malthouse, The value of online customer reviews, in: Proceedings of the 10th ACM Conference on Recommender Systems, ACM, 2016, pp. 155–158.
[2] Bettina von Helversen, Katarzyna Abramczuk, Wiesław Kopeć, Radoslaw Nielek, Influence of consumer reviews on online purchasing decisions in older and younger adults, Decis. Support Syst. 113 (2018) 1–10.
[3] New Data: 97% of Consumers Depend on Reviews for Purchase Decisions, https://www.powerreviews.com/events/consumers-depend-on-reviews/.
[4] Asad Abdi, Siti Mariyam Shamsuddin, Shafaatunnur Hasan, Jalil Piran, Deep learning-based sentiment classification of evaluative text based on multi-feature fusion, Inf. Process. Manage. 56 (4) (2019) 1245–1259.
[5] Abinash Tripathy, Abhishek Anand, Santanu Kumar Rath, Document-level sentiment classification using hybrid machine learning approach, Knowl. Inf. Syst. 53 (3) (2017) 805–831.
[6] Syed Muhammad Ali, Zeinab Noorian, Ebrahim Bagheri, Chen Ding, Feras Al-Obeidat, Topic and sentiment aware microblog summarization for Twitter, J. Intell. Inf. Syst. (2018) 1–28.
[7] Wenjuan Luo, Fuzhen Zhuang, Weizhong Zhao, Qing He, Zhongzhi Shi, QPLSA: Utilizing quad-tuples for aspect identification and rating, Inf. Process. Manage. 51 (1) (2015) 25–41.
[8] Ehsan Asgarian, Mohsen Kahani, Shahla Sharifi, The impact of sentiment features on the sentiment polarity classification in Persian reviews, Cognit. Comput. 10 (1) (2018) 117–135.
[9] Fatemeh Pourgholamali, Mohsen Kahani, Ebrahim Bagheri, Zeinab Noorian, Embedding unstructured side information in product recommendation, Electron. Commer. Res. Appl. 25 (1) (2017) 70–85.
[10] Charu C. Aggarwal, et al., Recommender Systems, Springer, 2016.
[11] Samaneh Moghaddam, Martin Ester, The FLDA model for aspect-based opinion mining: addressing the cold start problem, in: Proceedings of the 22nd International Conference on World Wide Web, ACM, 2013, pp. 909–918.
[12] Dae Hoon Park, Hyun Duk Kim, ChengXiang Zhai, Lifan Guo, Retrieval of relevant opinion sentences for new products, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 393–402.
[13] Md Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C. Van Esesn, Abdul A.S. Awwal, Vijayan K. Asari, The history began from alexnet: A comprehensive survey on deep learning approaches, 2018, ArXiv Preprint arXiv:1803.01164.
[14] Xing Guo, Shi-Chao Yin, Yi-Wen Zhang, Wei Li, Qiang He, Cold start recommendation based on attribute-fused singular value decomposition, IEEE Access 7 (2019) 11349–11359.
[15] Fatemeh Pourgholamali, Mining information for the cold-item problem, in: Proceedings of the 10th ACM Conference on Recommender Systems, in: RecSys '16, ACM, New York, NY, USA, 2016, pp. 451–454.
[16] Zhi-Peng Zhang, Yasuo Kudo, Tetsuya Murai, Yong-Gong Ren, Addressing complete new item cold-start recommendation: A niche item-based collaborative filtering via interrelationship mining, Appl. Sci. 9 (9) (2019) 1894.

[17] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, Zuoyin Tang, Collaborative filtering and deep learning based recommendation system for cold start items, Expert Syst. Appl. 69 (2017) 29–39.
[18] Y. Koren, Collaborative filtering with temporal dynamics, Commun. ACM 53 (4) (2010) 89–97.
[19] Yu Zhu, Jinghao Lin, Shibi He, Beidou Wang, Ziyu Guan, Haifeng Liu, Deng Cai, Addressing the item cold-start problem by attribute-driven active learning, IEEE Trans. Knowl. Data Eng. (2019).
[20] Chuan Shi, Binbin Hu, Wayne Xin Zhao, S. Yu Philip, Heterogeneous information network embedding for recommendation, IEEE Trans. Knowl. Data Eng. 31 (2) (2018) 357–370.
[21] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, Chi Xu, Recurrent knowledge graph embedding for effective recommendation, in: Proceedings of the 12th ACM Conference on Recommender Systems, ACM, 2018, pp. 297–305.
[22] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, Sen Wang, Learning graph-based poi embedding for location-based recommendation, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, ACM, 2016, pp. 15–24.
[23] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, Wei-Ying Ma, Collaborative knowledge base embedding for recommender systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 353–362.
[24] Xin Li, Hsinchun Chen, Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach, Decis. Support Syst. 54 (2) (2013) 880–890.
[25] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, Qiaozhu Mei, Line: Large-scale information network embedding, in: Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.
[26] Binbin Hu, Chuan Shi, Wayne Xin Zhao, Philip S. Yu, Leveraging meta-path based context for top-n recommendation with a neural co-attention model, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 1531–1540.
[27] Yinfei Yang, Cen Chen, Minghui Qiu, Forrest Bao, Aspect extraction from product reviews using category hierarchy information, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, 2017, pp. 675–680.
[28] Jay M. Ponte, W. Bruce Croft, A language modeling approach to information retrieval, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1998, pp. 275–281.
[29] Dragomir R. Radev, Hongyan Jing, Małgorzata Styś, Daniel Tam, Centroid-based summarization of multiple documents, Inf. Process. Manage. 40 (6) (2004) 919–938.
[30] Rada Mihalcea, Paul Tarau, Textrank: Bringing order into text, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), 2004, pp. 404–411.
[31] Chirantana Mallick, Ajit Kumar Das, Madhurima Dutta, Asit Kumar Das, Apurba Sarkar, Graph-based text summarization using modified textrank, in: Soft Computing in Data Analytics, Springer, 2019, pp. 137–146.
[32] W.X. Zhao, S. Li, Y. He, E.Y. Chang, J.R Wen, X. Li, Connecting social media to E-commerce: Cold-start product recommendation using microblogging information, IEEE Trans. Knowl. Data Eng. 28 (5) (2016) 1147–1159.
[33] Mengwen Liu, Yi Fang, Alexander G. Choulos, Dae Hoon Park, Xiaohua Hu, Product review summarization through question retrieval and diversification, Inform. Retrieval J. 20 (6) (2017) 575–605.
[34] Xiangnan He, Tao Chen, Min-Yen Kan, Xiao Chen, Trirank: Review-aware explainable recommendation by modeling aspects, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 1661–1670.
[35] Rose Catherine, William Cohen, Transnets: Learning to transform for recommendation, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, ACM, 2017, pp. 288–296.
[36] Xinyu Guan, Zhiyong Cheng, Xiangnan He, Yongfeng Zhang, Zhibo Zhu, Qinke Peng, Tat-Seng Chua, Attentive aspect modeling for review-aware recommendation, ACM Trans. Inform. Syst. 37 (3) (2019) 28.
[37] Yongfeng Zhang, Qingyao Ai, Xu Chen, W. Bruce Croft, Joint representation learning for top-n recommendation with heterogeneous information sources, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, 2017, pp. 1449–1458.
[38] Julian McAuley, Jure Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, 2013, pp. 165–172.
[39] Yunzhi Tan, Min Zhang, Yiqun Liu, Shaoping Ma, Rating-boosted latent topics: Understanding users and items with ratings and reviews, in: IJCAI, Vol. 16, 2016, pp. 2640–2646.
[40] Ding Xiao, Yugang Ji, Yitong Li, Fuzhen Zhuang, Chuan Shi, Coupled matrix factorization and topic modeling for aspect mining, Inf. Process. Manage. 54 (6) (2018) 861–873.
[41] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, Efficient estimation of word representations in vector space, 2013, ArXiv Preprint arXiv:1301.3781.
[42] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems (NIPS 2013), 2013, pp. 3111–3119.
[43] Mo Yu, Matthew Gormley, Mark Dredze, Factor-based compositional embedding models, in: NIPS Workshop on Learning Semantics, 2014, pp. 95–101.
[44] Taeuk Kim, Jihun Choi, Daniel Edmiston, Sanghwan Bae, Sang-goo Lee, Dynamic compositionality in recursive neural networks with structure-aware tag representations, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 6594–6601.
[45] Quoc Le, Tomas Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, 2014, pp. 1188–1196.
[46] N. Moratanch, S. Chitrakala, A survey on extractive text summarization, in: Computer, Communication and Signal Processing (ICCCSP), 2017 International Conference on, IEEE, 2017, pp. 1–6.
[47] You Ouyang, Wenjie Li, Sujian Li, Qin Lu, Applying regression models to query-focused multi-document summarization, Inf. Process. Manage. 47 (2) (2011) 227–237.
[48] Annie Louis, Ani Nenkova, Automatically assessing machine summary content without a gold standard, Comput. Linguist. 39 (2) (2013) 267–300.
[49] Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, Ani Nenkova, An assessment of the accuracy of automatic evaluation in summarization, in: Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization, Association for Computational Linguistics, 2012, pp. 1–9.
[50] Adam Berger, John Lafferty, Information retrieval as statistical translation, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Berkeley, CA, USA, 1999, pp. 222–229.
[51] Sung Jin Kim, Sang Ho Lee, An improved computation of the pagerank algorithm, in: European Conference on Information Retrieval, Berlin, Heidelberg, 2002, pp. 73–85.
[52] Toxic Comment Classification Challenge, https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion/52644.