Electronic Commerce Research and Applications 40 (2020) 100917

Contents lists available at ScienceDirect

Electronic Commerce Research and Applications

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

A neural graph embedding approach for selecting review sentences

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

[a] Department of Computer Engineering, Ferdowsi University of Mashhad, Iran
[b] Laboratory for Systems, Software and Semantics (LS[3]), Ryerson University, Canada

ARTICLE INFO

Keywords:
Cold-start
Graph-based information retrieval
Neural embeddings
Summarization

ABSTRACT

Product reviews written by the crowd on e-commerce shopping websites have become a critical information source for making purchasing decisions. An important challenge, however, is that the vast majority of products (e.g., 90% of products on amazon.com) do not receive enough attention and lack sufficient reviews by the users; hence, they constitute the so-called *cold products*. One solution to 1address cold products, which has already been studied in the literature, is to generate reviews for these products by sampling review sentences from closely related warm products. Our method proposed in this paper is specifically focused on such a solution. While a majority of the works in the literature rely on product specification similarity to identify relevant reviews that can be used for review sentence selection, our work differs in that it not only employs product specification similarity but also employs product-review, product-user, and user-review interactions when determining the suitability of a review sentence to be selected. More specifically, the contributions of our work can be enumerated as follows: (1) We propose that the selection of review sentences from other products should not only consider product-product similarity but also consider product-review, user-review, and user-user relationships. As such, we show how neural graph embeddings can be used to encode product, user, and review information into an attributed heterogeneous graph representation based on which similarities can be calculated. (2) We further propose how review *relevance* and *importance* can be considered using graph traversal to select appropriate review sentences for a given cold product. (3) Finally, we systematically compare the performance of our work with those of several state-of-the-art baselines on five datasets collected from CNET.com and rottentomatoes.com with different characteristics from both quantitative (e.g., the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics) and qualitative aspects and show how our proposed approach was able to provide statistically significantly improved performance over various strong baselines.

1. Introduction

Crowd-sourced product reviews play an important role in how customers make purchasing decisions in both online shopping sites and brick-and-mortar retail stores. A recent study has shown that 78% of U.S. digital shoppers bought a product at a physical store after reviewing it online (Overcoming). This is primarily because reviews allow users to increase their awareness and confidence in their purchasing decisions. Given the impact of product reviews on customer purchasing behavior, the research community has already extensively explored a host of computational methods that automatically analyze product reviews and mine actionable insight relevant for manufacturers, vendors, and end customers (Moghaddam and Ester, 2013). These methods cover both *descriptive analysis* of reviews, such as measuring review sentiments (Tripathy et al., 2017) and identifying product review aspects (Ali et al., 2018; Luo et al., 2015), and *predictive analysis* of products and customers including making effective product recommendations to users (Musat et al., 2013; Zhang et al., 2013; Zhang et al., 2017). While these methods prove very effective for popular products, they face challenges when working with cold-start products. In reality, many products offered in online shopping sites, including newly released products and, more generally, less popular products, receive very few, if any, reviews and are hence referred to as *cold products* (Pourgholamali, 2016).

To address this challenge, several leading product review websites, such as CNET, solicit product reviews from experts and present them to the end users. However, because cold products constitute close to 90% of products on many leading shopping websites such as amazon.com and epinions.com (Moghaddam and Ester, 2013), it becomes practically infeasible to curate manually written reviews for all such products. As a result, several studies have focused on the specific problem of generating product reviews that could then be quickly reviewed and edited

* Corresponding author at: Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran.
E-mail addresses: f.pourgholamali@mail.um.ac.ir (F. Pourgholamali), kahani@um.ac.ir (M. Kahani), bagheri@ryerson.ca (E. Bagheri).

https://doi.org/10.1016/j.elerap.2019.100917
Received 14 December 2018; Received in revised form 3 October 2019; Accepted 21 October 2019
Available online 16 December 2019
1567-4223/

by experts and posted on e-commerce websites (Park et al., 2015). The objective is to facilitate the process of providing reliable expert reviews on a product by offering an automated assessment of the product of interest and sketching an initial review of the product for the expert to consider. The foundation of existing works lies on the hypothesis that the greater the similarity between two products, the more likely it would be for end users to develop similar perceptions for these products, hence leading to the similarity of customer reviews for the two products. On this basis, the objective of existing works in the literature has been to define effective product similarity measures that could identify products whose reviews could be transferred to the product of interest. The work by Park et al. (2015) can be seen as one of the prominent works in this area that propose to use the products' structured specifications to find product similarities and has shown strong performance for retrieving review sentences for cold products. However, while earlier studies have shown that the structured specifications of products can represent the similarity between products very well (Barjasteh et al., 2016; Pourgholamali et al., 2017), one of the limitations of such approaches that focus on product specifications for similarity calculations is that not all products have clearly categorized specifications and, therefore, the similarity between products cannot be measured in such cases. Our proposed method builds on this limitation by proposing that the similarity between two or more products is a function of the interplay of many sources of information including product specifications, user-product engagement, and product-brand interaction. For this reason, an effective product similarity measure needs to holistically consider these sources of information within a unified framework. In this paper, we will show how several different types of information can be systematically considered to measure product similarities for the purpose of selecting and retrieving review sentences.

1.1. Research objectives and contributions

The objective of our proposed method is to curate product reviews for a given product by selectively sampling review sentences from other products that have already received genuine reviews from users. The core of our work is based on identifying product relationships that would determine which product review sentences should be selected for the product of interest. To identify product relations, our work models all products within an *attributed heterogeneous graph* representation, which would provide the basis for product similarity measurement and sentence selection. The graph representation is derived from the interplay of different sources of information related to products, users, and reviews. More concretely, we propose to compute unique product representations within a neural embedding space (Liao et al., 2018), which could then be used to measure product relationships. To compute neural embedding-based product representations, we initially model product, user, and review interactions within an attributed graph whose nodes are then transformed into a low-dimensional space while preserving the structural properties of the graph including the node interrelationships. The transformation of the attributed graph into a neural embedding space enables node similarity computation based on vector similarity computation. The advantage of the attributed heterogeneous graph representation of products is that it allows us to consider not only product specifications but also user-product interaction, user-review generation, and product-review relations when determining product relationships.

The embedding-based representation of the attributed graph allows us to measure product-review relations that have not been explicitly observed but are rather implicitly derived from the graph embedding. On the basis of these derived relationships and given a product of interest, we formulate a network consisting of products and reviews that are most relevant to the product of interest and perform a random walk traversal of the network to identify and select the most relevant review sentences. More formally, the main contributions of our work can be enumerated as follows:

1. We propose to model products, users, and reviews within an attributed heterogeneous graph representation such that the interaction between products, users, and reviews can be effectively captured when computing product relationships. We additionally show that the neural embedding of this attributed heterogeneous graph provides a dense vector representation of products, which can be used for product relationship calculation using simple vector similarity computation.
2. We further show that implicit product and review relations can be determined, even if not explicitly observed, on the basis of the neural embedding of the products and reviews from the attributed heterogeneous graph. We demonstrate that these implicit relations allow for the construction of a summarization network, the systematic traversal of which can be used to identify effective review sentences.
3. We performed extensive experiments on five real-world datasets collected from the CNET.com website and compared our method with state-of-the-art methods and show that our proposed method outperformed existing methods on a host of standard measures including the ROUGE-2 metrics.

The rest of this paper is organized as follows. Section 2 reviews the most relevant works related to the current work. Section 3 presents an overview of the proposed approach in a schematic form, which is followed by a detailed presentation of our proposed approach in Section 4. The details of our experimental setup, benchmark datasets, and baselines are presented in Section 5. Section 6 includes the analysis of our evaluation results. This section also offers a qualitative discussion of the performance of our method compared with those of other existing state-of-the-art methods. The paper is then concluded with final remarks and hints on future work in Section 7. We also provide a qualitative discussion on the output of different variations of our work as well as the outputs of the strongest baselines on different datasets in Appendix A.

2. Related works

The main focus of our proposed method is to procure additional review contents for cold products that do not have any or sufficient reviews. We will review how different authors have addressed the cold-start problem and then, more specifically, the methods that have offered solutions for the retrieval of relevant review sentences for online products. These methods basically retrieve review sentences that are most important and are related to the cold product. Among other methods, the problem of selecting review sentences for cold products can also be seen as a text summarization task, and as such, we will briefly review the related literature on automatic text summarization as well.

2.1. The cold-start problem

Zhao et al. (2016) were one of the first to propose that cold products could be handled by looking at the purchase transactional history of these products, at the purchase history of the rest of the products, and at the purchase history of the users. On this basis, the authors proposed to formulate a context for users and products by injecting their purchase history into a skip-gram neural embedding model such that neural representations can be developed for users and products within the same space. Alternatively, Pourgholamali et al. (2017) also proposed to create neural embeddings of products and users within the same embedding space; however, instead of using historical transaction history, they suggested incorporating unstructured side information such as reviews and product descriptions for building the embeddings. The developed embeddings not only allow for user-user and product-product similarity calculations but also enable user-product similarity

measurement. While these two approaches use the standard neural embedding representation techniques proposed in Mikolov et al. (2013), there are other methods that develop more intricate neural representations for products and reviews.

The method of Zhang et al. (2016) learns three types of representations for products, namely, structural embedding of items from a knowledge base by adopting a network embedding method, textual representations by adopting a Bayesian stacked denoising auto-encoder, and visual representation by applying a Bayesian stacked convolutional auto-encoder. These authors then proposed a Collaborative Knowledge base Embedding technique to jointly learn three representations in a unified model. Similarly but somewhat more limited in scope, a deep neural network architecture, called Deep-CoNN, was proposed by Zheng et al. (2017) for jointly modeling users' behavior and product properties using textual reviews. Users' behavior and the textual reviews are connected to each other through a shared layer that is optimized to minimize the distance between the textual reviews of a given product and the behavior of the users who interacted with that product. In a different work, Zhang et al. (2018) benefit from graph embedding techniques to incorporate social network information to address the cold-start problem. The authors introduce the concept of social network embedding in social information networks to generate user-specific features based on the network characteristics of the social network. The features are then incorporated into a matrix factorization model which can learn user-product and social features simultaneously. Finally, the authors interpolate the user-product and social features to the user-product ratings.

There have also been recent works that focus on building a multi-dimensional representation of products and users from different perspectives without embedding them in a lower-dimensional space. For instance, Zhang et al. (2017) proposed to construct a multi-viewed representation of users and products in the top k recommendation tasks. Each view is built on the basis of representing one type of information source, e.g., ratings, reviews, and images. The overarching representation of users and products is built by integrating the corresponding representations of users and products in each of these three views. Also systematically, Shi et al. (2018) propose a model based on Heterogeneous Information Networks (HIN). They consider using the concept of meta-paths (Sun et al., 2011) in a heterogeneous network to generate node sequences based on a random walk traversal defined over predefined meta-paths. For each meta-path, the authors learn a unique representation for products and users. The product and user representations obtained from the random walk are then incorporated into a matrix factorization model to provide recommendations for users.

Some authors have viewed the cold-start problem as a long-tail distribution of product ratings. Qiu et al. (2018) proposed a non-Gaussian embedding-based model to address this long-tail distribution. They converted the problem into a link prediction task for a bipartite graph (one part corresponds to users and the other corresponds to products) with multi-typed edges, where each type corresponds to a rating, ranging from 1 to 5. These authors then proposed a neural network that builds representations for users, products, and ratings. The representations of ratings were built on the basis of the translation embedding model that represents each rating as a translation between users and items in a rating-dependent subspace. Given a user and a product pair, the rating that can make a better translation between the user and the product is adopted as the correct rating.

The problem of cold-start products has also been investigated from the perspective of opinion mining techniques. For instance, and to address this problem, Moghaddam and Ester (2013) proposed the factorized LDA model, which is a probabilistic graphical model based on latent Dirichlet allocation (LDA) and addresses the problem of identifying aspects and estimating their ratings for cold products. This model assumes that both products and users can be modeled by a set of latent factors. Product factors represent the product's probability distribution over aspects and for each aspect its distribution over ratings. In the same way, user factors represent the user's probability distribution over aspects and for each aspect its distribution over ratings. These distributions are trained using the reviews of all the products of a category, in particular, the non-cold products, and serve as the prior for the distributions of cold products. For cold products, the aspect distribution is mainly determined by the prior aspect distribution of the category and the rating distribution of an aspect is mainly determined by the rating distribution of the user or by the prior rating distribution of all users (if the user is also cold, i.e., has written only a few reviews). Yang et al. (2017) later extended FLDA to take the hierarchy of the product category into account. This approach, known as CAT-LDA, models products in both general and specific categories.

As mentioned earlier, the problem of cold products has motivated the need to select relevant review sentences for them. The closest works to the theme of our paper are those of Liu et al. (2017) and Park et al. (2015), who have proposed probabilistic approaches to retrieve relevant review sentences from the non-cold-start yet similar products. These authors modeled the problem as a generative model that expects to estimate the specification of a cold product given a set of candidate review sentences. The hypothesis behind these works is that products with similar specifications have a higher likelihood of receiving similar reviews. There are some drawbacks to these approaches that we address in this paper, such as the following. 1) These approaches are primarily dependent on the products' structured specifications provided by the vendor or the manufacturer. For this reason, the models will not be able to provide any reviews for products that do not come with structured specifications. 2) The approaches overlook the role of other information sources such as user-product and user-review interactions when selecting review sentences for cold products. Moreover, 3) given that the generative models are based on a translation model, they are sensitive to the common words between product specifications and review sentences. This sensitivity to common words can impact the retrieval of semantically important review sentences. Our proposed method moves beyond the mere similarity of products based on their specifications and captures additional sources of information such as the relations between products and reviews, as well as the relations between users and reviews. Additionally, we model product-review interaction on the basis of an implicit notion of relevance derived from the embedding of product, user, and review information into an attributed heterogeneous graph, which would allow us to retrieve review sentences that not only are related to the product of interest but also are of semantic significance.

2.2. Automatic text summarization

As mentioned earlier, to address the cold-start problem in product reviews, one might define the problem of review sentence selection as an automatic text summarization problem. Therefore, we provide an overview of the pertinent literature on automatic text summarization in this section. Automatic text summarization is a technique that shortens long pieces of text to create a summary document that covers the major points of the original document. Most summarization systems are extractive summarizers, which identify the most salient sentences of the original document as the summary, as opposed to abstractive summarizers, which generate a new yet shorter textual snippet to convey the most critical information from the original document (Erkan and Radev, 2004). In general, extractive summarization methods can be categorized into two types: i) *topic representation approaches* and ii) *indicator representation approaches* (Nenkova and McKeown, 2012). In the topic representation approaches, the importance score of each sentence represents how well the sentence explains the most important topics of the input text. Indicator representation approaches represent each sentence in the input as a list of indicators of importance such as length, location in the document, and presence of certain phrases.

i) Topic Representation Approaches: Various techniques have been

adopted to identify the most important topics of the input document. Topic-word technique, frequency-driven approach, latent semantic analysis (LSA), and LDA are among them. In Dunning (1993), a topic-word approach was proposed and a log-likelihood ratio test was used to distinguish informative words that are referred to as *topic signature*. Then, a sentence score is computed either as a function of the number of topic signatures or as a proportion of the topic signatures in the sentence. Centroid-based summarization is a common topic-based representation baseline, which is based on the term frequency-inverse document frequency (TF-IDF) topic representation approach (Salton and Buckley, 1988). Furthermore, in Gong and Liu (2001) proposed to use LSA to select silent sentences in single- and multi-document summarization. At first, a term-sentence matrix (A) is built from the input text in which each row corresponds to a word and each column denotes a sentence. Each element of the matrix a_{ij} is computed by the TF-IDF weight of word i in sentence j. Then, using singular value decomposition (SVD), LSA decomposes matrix A into three matrices: $A = UWV^{\top}$. Matrix U is a term-topic matrix and $V^{\top}$ is a topic-sentence matrix. W is a diagonal matrix, and each row shows the weight of the corresponding topic. $WV^{\top}$ represents the weights of the topics in each sentence. Probabilistic topic models have gained attention in recent years (Na et al., 2014; Wang et al., 2009). For instance, a Bayesian sentence-based topic model for both single- and multi-document summarization is proposed by Wang et al. (2009).

ii) Indicator Representation Approaches: These methods try to build a representation of the input based on a set of features and assign scores to sentences on the basis of these features. Scoring sentences can be achieved either by graph-based approaches or by machine learning techniques. Commonly, in the graph-based approach, each sentence denotes a node and the edges between sentences show the similarity between them. These methods are usually based on a random walk model to identify silent sentences. More details on this summarization model that was adopted for the sentence selection in our proposed method are elaborated on in Section 4.2.1. On the other hand, machine learning methods model the summarization task as a classification problem that classifies input sentences as summary or non-summary sentences. Naive Bayes classifier, support vector machines, hidden Markov models, and conditional random fields are among the most common machine learning techniques used for the summarization task (Ouyang et al., 2011; Zhou et al., 2004). These methods, however, need labeled training data, which are often not available in all domains. Some approaches for curating annotated data (Ulrich et al., 2008) or for building semi-supervised models (Wong et al., 2008) have already been proposed to address this issue.

3. Approach overview

Given a specific product, the goal of our proposed approach is to select sentences from existing reviews of other products such that those sentences are highly related to the product of interest and cover the main aspects of the product that actual users may comment on. To this end, we propose a two-phase approach, as shown in Fig. 1, which consists of first building and embedding an attributed heterogeneous graph from user, product, and review information in the first phase and then subsequently exploiting the derived embeddings to construct a product network that would be used to select product review sentences for a given product of interest in the second phase.

As elaborated in more detail later, the first phase of our proposed approach consists of three steps. As we intend to represent product, user, and review information in the form of an attributed graph, initially, both textual attributes (e.g., short product summaries) and non-textual attributes (e.g., structured product specifications such as weight and dimensions) are encoded in a standard form for incorporation into the graph. Once the product attributes are encoded, we construct an attributed heterogeneous graph, which would consist of three node types representing the users, reviews, and products. The attributes on the product nodes would be the embedded attributes from the first step. Finally, this graph is embedded into a low-dimensional space where each node would be represented through an n-dimensional vector, providing an intuitive calculation of the similarity between graph nodes through vector similarity.

While the first phase of our proposed approach is generic and applies to the whole network of products, reviews, and users, the second phase is specific to a product of interest. As such, given the derived embeddings for products, reviews, and users, we construct a summarization network consisting of product and review sentence nodes such that the links in the network are determined on the basis of the vector similarity of the nodes from the embeddings. Having this derived network and one product of interest, we perform a random walk on the graph that would help identify those review sentences that must to be selected.

The details of the two phases and the proposed steps in each phase are presented in the following sections.

4. Proposed approach for review sentence selection

In the following subsections, we will provide elaborated details of the proposed two-phase approach for review sentence selection based on graph embeddings. The two phases and the steps involved will follow the flowchart outlined in Fig. 1. We should note that all the textual inputs are undergo a preprocessing step, where we remove stop words, tokenize the words, split the sentences, stem the words, and extract the review sentiments using the Natural Language Toolkit (NLTK) for Python.[1]

4.1. Graph embedding

The initial phase of our work is focused on learning an effective representation for products and reviews that can then be used for measuring the relevance between reviews and products. Intuitively speaking, products, users, and reviews constitute an interwoven network of interactions where users and products are connected to each other through browsing, purchasing, and reviewing activities. This network of interactions can be easily represented as a heterogeneous graph whose nodes are instances of products, users, and reviews, and whose edges form the interaction between each of these nodes. Now, on the basis of this graphical representation, we are interested in learning a compact representation for each node such that it would preserve the structural characteristics of each node within the graph, maintain the whole graph properties, and, at the same time, be simple and effective to use in our context. For this purpose, graph embedding techniques are suitable methods for this objective as they can transform the nodes of a graph into a dense low-dimensional vector representation such that all distance and geometric properties of the original graph are maintained. In our work, we are not only interested in building graph embeddings based on the structural properties of nodes in the graph, but also by considering node attributes, which can additionally describe the nature of each node. More concretely, we want the learned graph embedding to be cognizant of the relationship between the user, product, and review nodes, and, at the same time, also consider product attributes such as the weight, dimensions, and color when learning the embeddings. As such, considering products, users, and reviews as entities, our graph representation is in the form of an attributed heterogeneous graph that represents those entities as nodes of the graph. The interactions between entities constitute the edges of the graph, and the attributes of the entities form the node attributes. More formally, our representation graph can be defined as follows.

Given a set of reviews, users, and products denoted by R, U, and P, respectively, our representation graph $G = (G_{RU} \cup G_{RP})$ is a

[1] https://www.nltk.org/

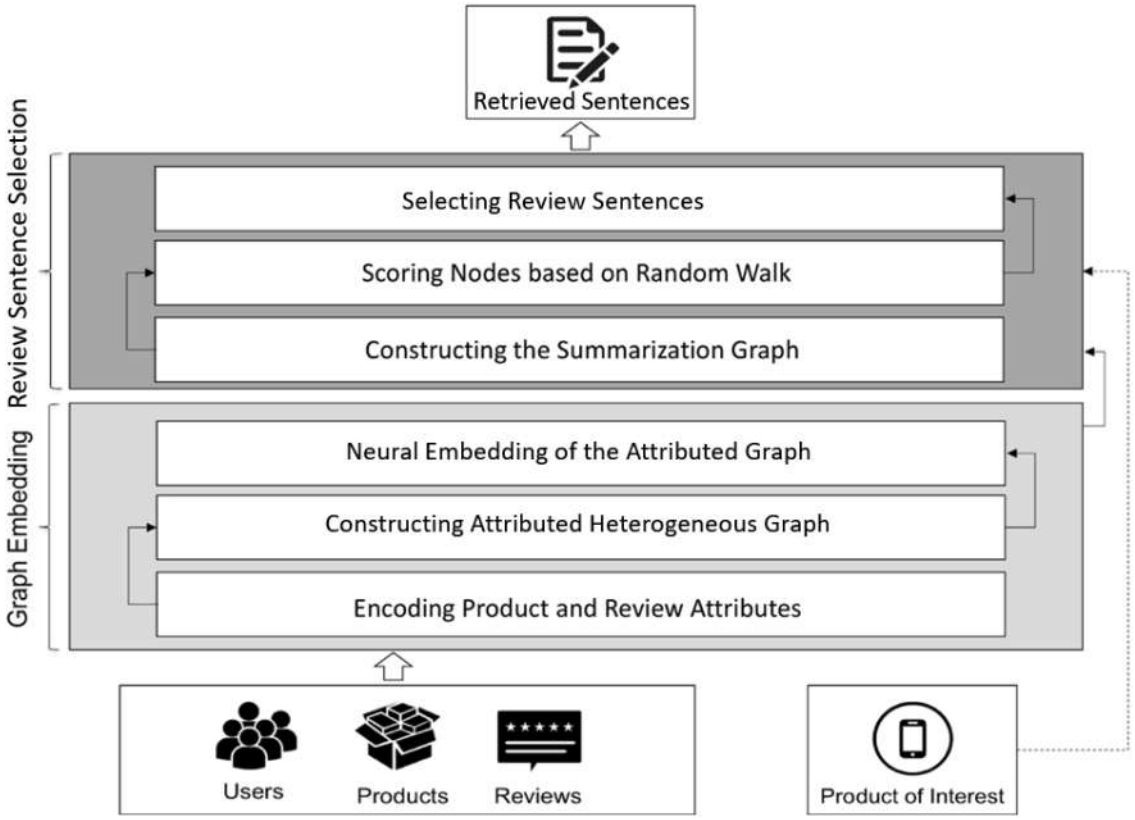

Fig. 1. The overview of our proposed approach.

heterogeneous graph composed of G_{RU} and . The subgraph G_{RP} denotes the product-review relationship in the form of $G_{RP} = (V_{RP}, E_{RP}, A_R \cup A_P)$, where G_{RP} is an unweighted attributed graph in which $V_{RP} = (V_R \cup V_P)$ denotes the union of the R and P nodes; E_{RP} represents the edges between products and reviews, denoting which review belongs to which product; and A_R and A_P represent the attribute sets of the reviews and products, respectively. Similarly, $G_{RU} = (V_{RU}, E_{RU}, A_R)$ is an unweighted attributed graph in which $V_{RU} = (V_R \cup V_U)$ denotes the union of user and review nodes; E_{RU} represents the relationships between user and review nodes, denoting which review was posted by which user; and A_R represents the attribute related to each review node.

It should be noted that theoretically speaking, it is possible to include other subgraphs such as user-product interaction (i.e., viewing or purchasing behavior), user-user interaction (e.g., social relationships between users), or additional attributes, such as user attributes (e.g., age and gender); however, given that such information is typically not publicly accessible from an online shopping store, without loss of generality and for the sake of repeatability of our experiments, we resort to the current formalization. Fig. 2 shows a schematic overview of the proposed network structure. It is clear that, if any other side information such as user attributes or the relation between users is available, it could be easily added in this schema in the form of either node attributes or edges between nodes.

While the nodes of the graph can be directly instantiated per product, user, or review, the incorporation of the attributes into the graph nodes requires some additional processing, which we will detail in the following section.

4.1.1. Encoding product and review attributes

In principle, product attributes can be broadly classified into (1) *free-form textual attributes* such as a written review or a product overview description and (2) *structured attributes* such as product

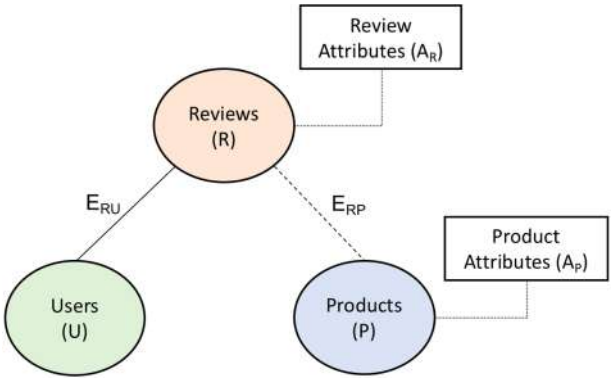

Fig. 2. The schematic overview of the attributed heterogeneous graph.

specifications. We systematically explain how each of these types of attributes can be incorporated into the graphical representation based on a standard feature vector formalism.

(1) Encoding Free-Form Textual Attributes: The most common approach to represent free-form textual content is to use a bag-of-words approach, where each observed word would become a feature and various measures of frequency such as TF-IDF and Best Match 25 would be employed to determine the feature value. It has been widely reported that the use of the bag-of-words approach leads to the curse of dimensionality owing to the large number of unique words that are observed in the vocabulary space. Alternatively, and in the context of reviews and product attributes, several studies (Musat et al., 2013; Poria et al., 2016; Saveski and Amin, ACM. 2014,; Xiao et al., 2018) have already shown that such free-form text can be viewed in the form of aspects. Aspects refer to specific characteristics of a product that are of potential interest to the users. As such, and among different approaches, topic modeling techniques such as LDA have been employed

on free-form product reviews and descriptions to identify possible aspects. On this basis, we propose to form a feature vector for free-form textual attributes of products and reviews by learning a topic model over the collection of the free-form textual corpus. This will produce a set of k topics, each of which will serve as one of the features. More formally, the feature vector representation of free-form textual content is defined as follows.

Given a collection of free-form textual content such as user reviews or product descriptions, a topic model learned on this collection will infer two distributions, namely, a topic-term distribution and a document-topic distribution. In the context of our work, assuming k topics derived by LDA, we represent each free-form attribute as a vector of weights $(w_1, w_2, ..., w_k)$, each element of which corresponds to the weight obtained by the document-topic distribution for a specific topic.

It is worth mentioning that our objective is to exploit auxiliary attributes to reveal the implicit relations between cold products and other entities. Therefore, our main focus is to encode any attributes that can facilitate this objective for us. For this reason, we additionally consider review sentiments represented in the form of review node attributes. The sentiment attributes can have three values corresponding to positive, negative, and neutral sentiments.

(2) Encoding Structured Attributes: Structured information on a shopping site can be broadly classified as numerical and categorical attributes. While the representation of numerical attributes is straightforward, where each attribute is represented as a feature, it is not recommended to use this representation strategy for categorical attributes. One of the common approaches for representing categorical attributes is to use one-hot encoding. However, in the context of our work, we find that using one-hot encoding can lose some important attribute information and, as such, might not be appropriate for our work. For instance, consider four different cameras that have the following values for their product-type attribute: (i) Digital camera – SLR with Live View mode, with Movie recording, (ii) Digital camera – SLR with Live View mode, (iii) Digital camera – SLR and (iv) Digital camera – Compact. In a one-hot encoding approach, each of these four values will be placed in a separate attribute slot; however, when we look at the attribute values, it is clear that the attribute of product (i) is much closer to product (ii) than to the other two products, which would not be captured if a one-hot encoding scheme is used. For this reason, we propose to use a bag-of-words approach for representing categorical attributes. For each attribute, we build a collection of words observed over all products for that attribute to form the vocabulary space. We then compute the TF-IDF of each word observed in a specific product to instantiate the feature vector of that attribute for the product. The reason why a bag-of-words representation can be appropriate in this context is that the number of words used over all products for a given categorical attribute is limited and does not present the curse-of-dimensionality challenge.

For each product and review node, we encode its free-form textual and structured attributes as discussed above and employ them to augment the nodes in the heterogeneous graph with attributes, hence leading to an attributed heterogeneous graph. The encoding schema of the review and product node attributes is illustrated in Fig. 3. We reserve the first k values of the feature vector for the textual attributes. The three subsequent values are reserved for the sentiments of the review nodes. The remaining vector is allocated to the structured attributes of the product nodes. We represent each value by a circle, and the color intensity denotes the high or low value of the feature. If a specific type of attribute is not available for a node type, such as textual data for user nodes, the corresponding values in the feature vectors are set to zero, as shown by the hollow circles in Fig. 3. All values in the feature vector of user nodes are set to zero since there is no side information available for them in our datasets.

4.1.2. Neural embedding of the attributed graph

The attributed heterogeneous graph incorporates the relationship between products and reviews, as well as between reviews and users, and also various types of attributes related to products and reviews. A transformation of the nodes within this graph representation into a dense vector representation will enable us to compute the similarity between the nodes of this graph even if these nodes are of different types. We are interested in a transformation that would guarantee the preservation of both structural proximity and attribute proximity of the nodes within the graph. A recent work in the literature that proposes systematic ways to perform this transformation, through neural embeddings of heterogeneous graphs, relies on the concept of meta-paths (Sun et al., 2011). A meta-path specifies the permissible paths between different node types to connect a source node type to a destination node type. As can be seen in the graph schema shown in Fig. 2, the underlying graph structure in our problem consists only of one meta-path and, hence, it would not be appropriate to apply neural embedding approaches based on meta-paths. In its lieu, we adopt a neural network-based architecture (Liao et al., 2018) to embed graph nodes into a vector representation by simultaneously considering the node relationship and attribute similarity. The architecture of the network is shown in Fig. 4 and comprises multiple layers. The input layer includes two inputs, namely, the one-hot encoded identifier of a node n from the graph, denoted as W^{id}, and the vector representation of the attributes of the same node, specified as W^{att}. The second layer consists of two components. In the first component, the one-hot encoded node n identifier input is mapped onto a dense vector representation (e) such that the structural relation between nodes is preserved. The second component also builds a dense representation (e') by aggregating the node attribute information. Now, the dense representations, namely, e and e', are fed into a hidden layer component with l hidden layers. This component is in the form of a tower structure where each layer is half of the size of the previous hidden layer. The first hidden layer is defined as $h^{(1)} = \begin{bmatrix} e \\ \lambda e' \end{bmatrix}$ where λ is a vector that adjusts the importance of the attributes. In our work, we adopt a unit vector to represent λ to give the attributes the same importance. Each hidden layer is then connected to the next through a feedforward architecture such that, at the k^{th} hidden layer, we have $h^{(k)} = \delta_k(W^{(k)}h^{(k-1)} + b^{(k)})$, where k is a number between 1 and l;$W^{(k)}$ and $b^{(k)}$ are the k^{th} hidden layer weight matrix and biases, respectively; and δ_k is an activation function. We adopt the soft-sign activation function since it has performed well in a similar setting (Liao et al., 2018). Finally, the output layer is a vector consisting of the

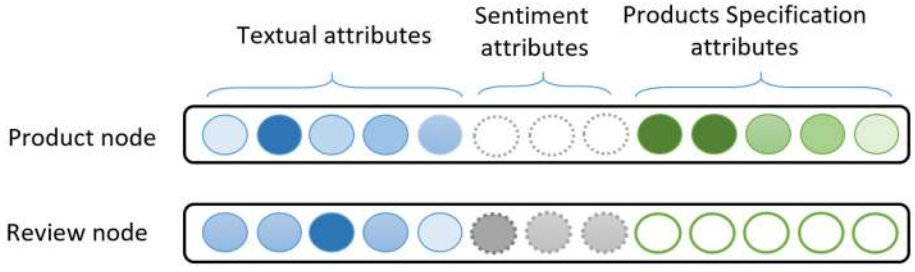

Fig. 3. The schematic of the feature vectors for review and product nodes.

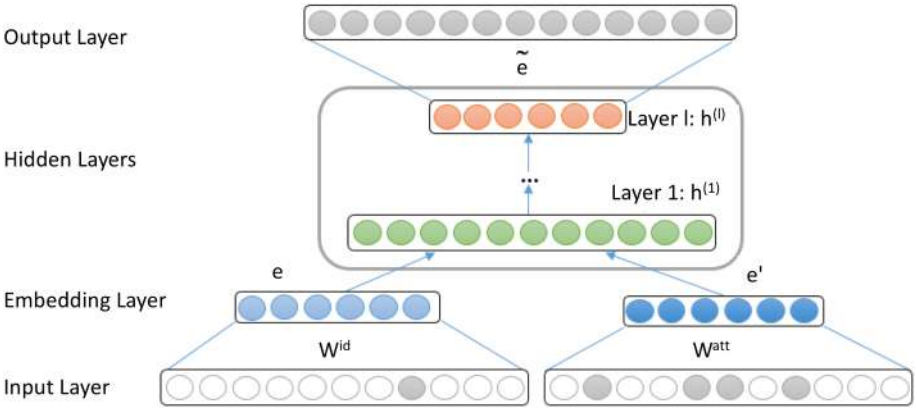

Fig. 4. The neural architecture used for learning the graph embeddings.

probability of linking the input node to each of the other nodes in the graph, denoted by $[p(n_1|n_i), p(n_2|n_i), ...,p(n_N|n_i)]$, assuming that the network comprises N nodes. Each element of this vector is computed on the basis of a softmax function as

$$p(n_j|n_i) = \frac{exp(\tilde{e}_j . h_i^{(l)})}{\sum_{j'=1}^{N} exp(\tilde{e}_{j'} . h_i^{(l)})} \tag{1}$$

where e_j is the vector representation learned for node j (n_j) and $h_i^{(l)}$ is the output vector from the last hidden layer for the current node n_i. Now, given the formalization of the structure of the output node, we learn the network parameters (Θ) by maximizing the conditional edge probability over all nodes in the graph. Therefore, we have

$$\begin{aligned}\Theta^* &= \text{argmax} \prod_{i=1}^{M} \prod_{j \in N_i} p(n_i|n_j) \\ &= \text{argmax} \sum_{n_i \in M} \sum_{n_j \in N_i} \log p(n_i|n_j)\end{aligned} \tag{2}$$

where Θ^* is the optimized network parameters, N_i denotes the set of node n_i's neighbors in the graph, and M is a random sampled set of entities from the set of all entities in N drawn for the purpose of negative sampling (Mikolov et al., 2013). Regarding the softmax schema in Eq. (1), we have

$$\Theta^* = \text{argmax} \sum_{n_i \in M} \sum_{n_j \in N_i} \log \frac{exp(\tilde{e}_j . h_i^{(l)})}{\sum_{j'=1}^{|M|} exp(\tilde{e}_{j'} . h_i^{(l)})} \tag{3}$$

It is worth mentioning that considering the edge probability in the objective function implies that we have considered the first order proximity in modeling the structure proximity in the graph embedding approach. When we consider the gradient of this log probability, we have

$$\nabla \log p(n_j|n_i) = \nabla \tilde{e}_j . h_i^{(l)} - \sum_{n_{j'} \in M} p(n_{j'}|n_i) \nabla \tilde{e}_{j'} . h_i^{(l)} \tag{4}$$

Liao et al. (2018) suggested using Kingma and Ba (2015) to optimize such a network structure. Adam applies smaller updates for the frequent parameters and larger updates for the infrequent parameters to obtain the learning rate and uses the dropout regularization method. Once the parameters of the proposed method are optimized on the basis of the conditional edge probability as defined above, each node will receive an embedding representation based on the value of the dimensions of the last hidden layer, $h^{(l)}$. It is now possible to compute the similarity of any two nodes in the attributed heterogeneous graph using the similarity of their learned embedding representation. Some authors such as (Feng et al., 2017; Sun et al., 2015; Zhao et al., 2016) have proposed that the cosine similarity between two embeddings is a useful measure of relatedness, which we adopted in our experiments.

4.1.3. Scalability analysis

Developing cost-efficient solutions for machine learning tasks is a critical issue that should be considered carefully. Neural networks require a large number of operations for efficiently learning their parameters. Although the training time of the graph embedding method was quite short in our experiments, to systematically study the scalability of the adopted graph embedding method, we will discuss its time complexity.

Regarding Fig. 3, we assume that the input vector can be described as $x + y$, where x is the one-hot encoding of the entities (nodes) and y is the attribute vector of the entities. If we consider the number of entities and the number of attributes as N and L, respectively, we have $x \in \{0, 1\}^N$ and $y \in \mathbb{R}^L$. In reference to Section 4.1.1, the attribute encoding of the entities is performed in such a way that the attribute vector has a small dimension. As mentioned earlier, when we use structured specifications, each numerical attribute would increase the dimension of the attribute vector (L) by 1. Categorical attributes that have small and distinct values are encoded by the TF-IDF weighted bag of words. Therefore, they would only slightly increase the dimension of the attribute vector. Textual attributes are also encoded by the number of topics defined in LDA. As a result, we can consider parameter L to be constant. The only parameter that can impact time complexity is N. Therefore, we proceed with the computations by considering N as the size of the problem.

In neural networks, the input is treated in the same way as an activation matrix in layer 0 ($a^{(0)}$). Therefore, we have $x = a^{(0)}$. The neural network adopted in this research has a feedforward architecture, and, therefore, in its k^{th} layer, we have $z^{(k)} = W^{(k)} a^{(k-1)}$, where $W^{(k)} \in \mathbb{R}^{n^{(k)} * n^{(k-1)}}$ and $n^{(k)}$ and $n^{(k-1)}$ are the dimensions of the $h^{(k)}$ and $h^{(k-1)}$ hidden layers, respectively. We also have $a^{(k)} = g(z^{(k)})$, where $g(x)$ is the activation function, which is evaluated in an element-wise way. We see that, for each layer, one matrix multiplication, as well as an activation function, is computed. Therefore, we have

$$time = n_{mul} + n_g \tag{5}$$

where n_{mul} is the number of multiplications performed and n_g is the number of times that the activation function is applied. More specifically, in layer k we have a matrix W:

$$W^{(k)} \in \begin{cases} \mathbb{R}^{n^{(k)}} & k = 0 \\ \mathbb{R}^{n^{(k)} * n^{(k-1)}} & k \neq 0 \end{cases} \tag{6}$$

now we have

$$n_{mul} = \sum_{k=2}^{n_{layers}} n^{(k)} * n^{(k-1)} * n^{(k-2)} + n^{(0)} * n^{(1)} \tag{7}$$

Let us assume that the number of neurons in each layer is equal to N (in fact, it is half of the size of the earlier hidden layer). With N, an optimal matrix multiplication operation would have a time complexity of $O(N^{2.37})$ (Coppersmith and Winograd, 1990). Since $g(x)$ is an element-wise function, we know that it has a running time of $O(N)$. Given that n_{layers} is a constant, we will have

$$O(N^{2.37}) + O(N) = O(N^{2.37}) \tag{8}$$

Therefore, the time complexity of training the network for a single node is $O(N^{2.37})$. Given that we have N nodes in the graph, the total time complexity would be $O(N^{3.37})$. We can see that the most time-consuming part of the graph embedding algorithm is the matrix multiplication. In practice, for large-scale matrices, it is easier and faster to use parallel computation over GPUs for matrix operations. There are several parallel algorithms that propose scalable and efficient solutions for matrix multiplication on distributed architectures. Geijn and Watts (1997) and Cannon (1969) are two examples of such parallel algorithms. Moreover, in recent years, the big data community has provided efficient frameworks for handling large-scale matrix data. Apache Spark is an open-source analytics framework for big data processing and contains built-in libraries for general-purpose cluster computing operations. It supports operations on distributed matrices in its MLlib module.[2] Therefore, while the time complexity of this embedding technique is $O(N^{3.37})$, in practice, it is easy to significantly scale it for large-scale networks.

4.2. Review selection

The proposed approach for embedding the nodes of the attributed heterogeneous graph into a dense vector space enables us to effectively compute node similarity even for nodes that are not structurally linked together in the graph. For example, the embeddings enable the comparison of any two products in such a way that not only product specifications are taken into account, but also the similarity of their reviews and the engaged users are considered. Now, on the basis of our ultimate objective, we need to find the most appropriate set of review sentences for a product. As such, we construct a second heterogeneous graph structure, referred to as the *summarization graph*, which is constructed on the basis of the node embeddings of the first attributed heterogeneous graph. The summarization graph consists of two node types, namely, products and review sentences, and three edge types representing product similarity, review sentence similarity, and product-review sentence relations. More formally, the summarization graph can be defined as follows.

Given a set of review sentences, each of which is derived from splitting reviews into sentences and products, denoted by *RS* and *P*, respectively, our summarization graph $G_s = (G_P \cup G_{RS} \cup G_{PRS})$ is a heterogeneous graph composed of G_P, G_{RS}, and G_{PRS}. The subgraph G_P denotes the product-product relationship in the form of $G_P = (V_P, E_P)$, which is an unweighted graph where V_P denotes the set of P nodes and E_P represents the edge set that connects the products with each other. There would be an edge between two products if the similarity of their embedding is greater than a threshold α.

Similarly, $G_{RS} = (V_{RS}, E_{RS})$ is an unweighted graph in which V_{RS} denotes the set of all review sentences observed across all reviews and E_{RS} represents the relationship between two review sentences if the cosine similarity of the two sentences is greater than a threshold β. Finally, $G_{PRS} = (V_P \cup V_{RS}, E_{PRS})$ is an unweighted graph where the nodes are the set of *P* and *RS*. We construct the edge set as follows: Given a sentence s in review r and a product p, there would be an edge between s and p if the similarity of r and p is greater than the threshold α. Fig. 5 shows a schematic overview of the heterogeneous summarization graph. It should be noted that, as the graph edges are unweighted, the graph structure is dependent on two hyperparameters, namely, α and β, which we will examine in our experiments.

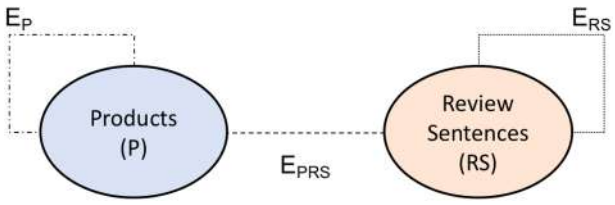

Fig. 5. The schematic overview of the summarization graph.

4.2.1. Scoring nodes on the basis of random walk

To select a set of appropriate review sentences for a product, we need to score and rank the nodes on the basis of their relevance. As we mentioned earlier, this task could be considered to be similar to an *extractive summarization* task (Moratanch and Chitrakala, 2017) since we aim to select a subset of the review sentences. Centrality-based summarization has shown to be one of the most popular and efficient summarization techniques in which sentences with high centrality are selected as the summary of the document (Radev et al., 2000; Radev et al., 2004). We pursue a similar objective where we intend to sample review sentences from G_s on the basis of a metric that would denote the centrality of review sentences based on the information encoded in the summarization graph. Centrality-based measures have been extensively studied in the context of graph-based summarization and have shown to have stronger performance than those of other state-of-the-art measures (Erkan and Radev, 2004; Mihalcea and Tarau, 2004). 2004,; Tan et al., 2017; Xiong and Ji, 2016). Some authors have proposed that the degree of each node can be considered as the centrality of the node in the graph (Erkan and Radev, 2004; Oya, 2015). However, degree centrality may havea negative effect on the quality of the retrieved sentences where a group of highly similar review sentences exist in the review collection, which reinforce each other, leading to high degree centralities, thus biasing the selected sentences. To address this issue, we opt to consider the centrality of adjacent nodes when computing the degree centrality of review sentence nodes. In other words, we consider that each node in the summarization graph has some centrality value and can distribute this centrality value to its neighbors. This formulation can be expressed as

$$p(n) = \sum_{n' \in N_n} \frac{p(n')}{deg(n')} \tag{9}$$

where $p(n)$ is the centrality of node n, N_n is the set of neighboring nodes of n, and $deg(n')$ is the degree of node n'. We can rewrite $p(n)$ for all nodes n in a matrix notation by dividing each value of the adjacency matrix by the degree of the corresponding node. Kim and Lee (2002) showned that such a matrix satisfies the properties of a stochastic matrix, which is a square matrix used to describe the transitions of a Markov chain. As such, the centrality vector would be the stationary distribution of the matrix. The stationary distribution at a node is related to the amount of time a random walker spends visiting that node and, hence, is a sign of centrality of that node. There has already been a proposal to assign some low probability for jumping between arbitrary nodes in the graph to avoid periodic or disconnected components, which makes the graph irreducible and aperiodic. As such each $p(n)$ can be reformulated as

$$p(n) = \frac{1-d}{N} + d \sum_{n' \in N_n} \frac{p(n')}{deg(n')} \tag{10}$$

where N is the total number of nodes in the summarization graph and d is a damping factor. We adopt such a random walk strategy over the summarization graph to produce centrality measures for each review sentence, which is equivalent to $p(n)$ for review sentence node n.

[2] https://spark.apache.org/docs/latest/mllib-data-types.html

4.2.2. Selecting review sentences

To identify and select review sentences that are both relevant to the product of interest and also important review sentences, we need two information sources revealing the *relevance* and *importance* of the review sentences. On the one hand, the relevance of a review sentence for a given product can be determined on the basis of the traversal of the edges between products and review sentence nodes in the summarization graph. On the other hand, the importance of a review sentence can be captured through the centrality score computed for each review sentence node in the previous section, i.e., $p(n)$. Given the summarization graph and the centrality score of each review sentence node in this graph, our objective is to traverse the summarization graph starting from the product of interest such that important and relevant reviews are reached. To this end, we propose a depth-first search (DFS)-based traversal technique for this task. For any given product, the summarization graph expresses that any node connected to the product of interest is already highly similar to this product and, therefore, is highly *relevant*. From among the highly relevant nodes and at each depth of the DFS traversal, we choose the neighbor with the highest degree of centrality, i.e., highest *importance*, to be included in the selected sentences for the product. Considering the transitive property of the similarities between nodes in the summarization graph, we sift through the summarization graph using a DFS strategy until a proper number of sentences are retrieved for the product. The reason a DFS algorithm was chosen is to avoid repetitive and highly similar review sentences from being selected. This is because DFS will diversify over different products at various depths, and, hence, it will avoid the selection of overly similar and repetitive review sentences.

5. Experimental setup

In this section, we introduce the datasets used to run our experiments as well as the metrics adopted to measure the performance of our approach against other baseline techniques. We also formally introduce the baseline techniques employed for comparing our work. We report our findings based on these datasets and metrics against the baselines in a subsequent section.

5.1. Datasets

We selected two domains to investigate our work. The first was inspired by the work of Park et al. (2015), who collected product information from the CNET website, which contains reviews and specifications for commercial products. The two product categories used by Park et al. were the Digital Camera and MP3 Player categories. The products and relevant information of these two categories were crawled from CNET for products available on February 22, 2012. We adopted these two datasets to perform our experiments. Moreover, we investigated and benchmarked our work on cold products. According to the definition of cold products proposed by Moghaddam and Ester (2013), any products with less than 10 reviews are considered to be cold, and, hence, we excluded any products that have more than 10 reviews. Products with less than 10 reviews formed two additional datasets. These datasets also contain reviews written by CNET editors, the textual description text of the product, and the structured specification of each product, which we have used for incorporating attributed information as outlined in Section 4.1.1. The second domain that we used was the movie domain. We selected a dataset from the Rotten Tomatoes website.[3] Rotten Tomatoes is a movie review aggregator database that also contains structured specifications for movies as well as a textual description for them. Both regular users and critics are allowed to review the movies. We crawled reviews, movie information 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. We selected movies with the documentary-based genres and their reviews from the critics as our fifth dataset. Since in this dataset, most of the movies had received rather a large number of reviews, we did not extract an additional dataset from the Dataset 5 as the cold version.

Table 1 outlines the details of the five datasets including the average number of reviews per product (RPP), average number of reviews per user (RPU), and average number of sentences per review (SPR). We will discuss the impact of these metrics on the performance of the various approaches in our findings.

5.2. Evaluation metrics

Park et al. (2015) proposed to use two sets of metrics for evaluating work in the area of sentence review selection. The first is based on the idea that an appropriate sentence set is one that has the closest similarity to an actual product review. As such, these authors proposed to compare the selected sentences with an actual review on the basis of the TF-IDF cosine similarity of the two reviews defined as follows:

$$TF-IDF-CosSim(D_1, D_2) = \frac{\sum_{w\in D_1, D_2} c(w, D_1).\, c(w, D_2).\, IDF(w)^2}{\sqrt{\sum_{w\in D_1} (c(w, D_1).\, IDF(w))^2}\,.\,\sqrt{\sum_{w\in D_2} (c(w, D_2).\, IDF(w))^2}} \tag{11}$$

Here, $c(w, D)$ represents the number of words w that occur in document D and the *IDF* of word w is defined as

$$IDF(w) = \log\frac{|R|}{1 + DF(w)} \tag{12}$$

where $|R|$ denotes the number of documents in the whole corpus and $DF(w)$ is the number of documents that contain w.

From a different perspective, given that our proposed sentence selection approach is a process that selectively samples review sentences from other products' reviews, it can also be viewed as a summarization process. As such, it is also appropriate to evaluate the selected sentences using standard document summarization metrics. One of the widely used metrics in document summarization, known as the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics, evaluates a generated summary in contrast with one or multiple human-generated reference summaries through the number of overlapping *n*-grams (Lin, 2004). In the context of our work, we assumed that the sentences retrieved by our proposed approach and by the other baselines were the generated summary and that the actual reviews of the product were the reference summaries. On this basis, we applied the ROUGE metrics for evaluating our work. As ROUGE-2 has already been reported by other researchers to be among the most reliable ROUGE metrics for evaluating the quality of a summary (Louis and Nenkova, 2013; Owczarzak et al., 2012), which is in essence based on bi-gram matching, we adopt ROUGE-2 in our evaluations, defined as follows:

$$ROUGE2-recall = \frac{\sum_{bigrams\in s} Count_{match}(bigram)}{\sum_{bigrams\in r} Count(bigram)} \tag{13}$$

$$ROUGE2-precision = \frac{\sum_{bigrams\in s} Count_{match}(bigram)}{\sum_{bigrams\in s} Count(bigram)} \tag{14}$$

where s and r denote system-retrieved sentences and reference reviews, respectively. $Count_{match}$ refers to the number of co-occurring bi-grams in s and r. When there are several reference summaries, ROUGE proposes to take the maximum from the reference summaries. Therefore, when computing both the ROUGE and the TF-IDF cosine similarity metric, we take the maximum between all r_i and s, where r_i is the i^{th} reference review.

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

Table 1
The Statistics of the Five Datasets Obtained from CNET.com and rottentomatoes.com.

Dataset Name	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5
Description	Digital Cameras in general	MP3 Players in general	Digital Cameras products with less than 10 reviews	MP3 Players products with less than 10 reviews	Movies, Documentary based genres, critics reviews)
Number of products	1,153	605	855	418	385
Number of users	7,506	5,735	2,394	1,060	1,326
Number of reviews	8,856	6,775	2,503	1,157	6,467
Number of sentences	80,980	119,208	23,888	19,515	7,466
Average number of reviews per product (RPP)	7.7	11.2	2.9	2.8	16.79
Average number of reviews per user (RPU)	1.17	1.18	1.04	1.09	4.88
Average number of sentences per review (SPR)	9.1	17.6	9.6	16.7	1.15

In the text summarization literature, reference summaries are typically those summaries that are created by human judges/experts. It has been shown that evaluating a summary using multiple human-generated summaries (which are also referred to as *models*) provides higher correlation with human judgments than the correlation obtained when evaluating a summary using a single model (Louis and Nenkova, 2013). However, in general, obtaining multiple reference summaries from human experts is not an easy task. There are also some studies in the literature that propose not to rely solely on human-generated summaries as the reference summaries to evaluate a system-generated summary. They suggest that another possible solution is to exploit the source text and evaluate the summary by computing the similarity between the source and the summary text. Several metrics of this type have been explored in Annie and Nenkova (2009). Pseudo-models are also proposed to augment the evaluation when there is only a single model available. At first, several automatic summarization systems are evaluated on the single model. Then, the output of the best system is regarded as the pseudo-model (Louis and Nenkova, 2013). Another solution is the consensus-based evaluation method, which is proposed for situations when there is no model available. This metric utilizes the output of some high-performing automatic summarization system for performing quality assessment on the system under evaluation (Louis and Nenkova, 2013). It is worth noting that none of these evaluation approaches is applicable in our task. We cannot use a metric that relies on the source document since there is no source document available for our task. Pseudo-models and consensus-based metrics are not applicable either because there are no high-quality automatic systems that retrieve review sentences or generate reviews for cold products that can adopt them as references.

5.3. Baseline techniques

There are a few works in the literature that have addressed the problem of selecting and retrieving relevant sentences for e-commerce products. We used these works as the baselines in our experimental evaluation. We briefly introduce each of these methods and how they are implemented for the sake of reproducibility. For better readability, our employed notations are summarized in Table 2 and an overview of the baselines is summarized in Table 3.

QL: For the first baseline, we examine a standard ad hoc retrieval method, the query likelihood (QL) language model approach (Ponte and Bruce Croft, 1998). The score function in QL is defined as Park et al. (2015)

$$score(t;R, S_z) = \sum_{k=1}^{F} \prod_{w \in s_{z,k}} p(w|t) \tag{15}$$

where $s_{z,k}$ is the set of words in the k^{th} feature-value pair in S_z. In this formulation, the specification words of cold products are evaluated by $p(w|t)$, which follows the unigram language model (Ponte and Bruce Croft, 1998).

MEAD: MEAD is a standard centroid-based multi-document summarization technique proposed in Radev et al. (2000). On the basis of this method, we sort review sentences according to their centroid scores using the following formula:

$$score(t;R) = w_c C_t + w_o O_t \tag{16}$$

where C_t is the sum of the centroid scores of the words in t and O_t is a position score, which gives higher scores to the review sentences appearing earlier in a document. The centroid score of a word is its TF-IDF value in the corpus R, and w_c and w_o are the weights for C_t and O_t, respectively. O_t is defined as

$$O_t = \frac{(n - i + 1)}{n}. C_{max} \tag{17}$$

where given the list of sentences in the document, n is the length of this list, i is the index of the sentence t in the list, and C_{max} is the maximum centroid score of the sentences in the document.

MEAD-SIM: We also adopted the modified version of MEAD, called MEAD-SIM, proposed in Park et al. (2015) as another baseline, which computes sentence scores on the basis of both the centrality of sentence t and the proximity of P_y and P_z, as follows:

$$score(t, S_y;R, S_z) = C_t. SIM_p(P_y, P_z) \tag{18}$$

Here, $SIM_p(P_y, P_z)$ defines the similarity between two products P_y and P_z on the basis of the cosine similarity of their structured specifications.

RevSpecGen: This baseline is built on the basis of a query likelihood model (Berger and Lafferty, 1999), which assumes that a document generates a query. As such, this approach aims to generate a specification of the product (S_z) from a candidate sentence t via a generative process: A candidate sentence t of product P_y generates R_y^{-t}, which refers to reviews for product y except for t. Then, t and R_y^{-t} are used to jointly generate the specifications for y, denoted as S_y. S_y then generates the query specification S_z.

$$\begin{aligned} score(t, R_y^{-t}, S_y;R, S_z) &\propto p(t, R_y^{-t}, S_y|S_z) \\ &= \frac{p(S_z \mid S_y)p(S_y \mid t, R_y^{-t})p(R_y^{-t} \mid t)p(t)}{p(S_z)} \\ &\propto p(S_z|S_y)p(S_y|t, R_y^{-t})p(R_y^{-t}|t) \end{aligned} \tag{19}$$

Simply put, in this scenario, t is scored on the basis of the similarity

Table 2
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

Table 3
Descriptions of the Baseline Approaches.

Method	Citation	Description
QL	Ponte and Bruce Croft (1998)	Query likelihood language model retrieval, which scores the sentence t on the basis of the similarity between t and the specifications of P_z.
MEAD	Radev et al. (2000)	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	Park et al. (2015)	Modified version of MEAD, which scores a review sentence on the basis of its centrality and of the similarity between products.
RevSpecGen	Park et al. (2015)	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. (2015)	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 translations of the review. The selected review and each of the generated review sets jointly generate a possible specification for a relevant product related to the selected review.

between products P_z and P_y; the similarity between P_y, t, and R_y^{-t}, which is computed on the basis of the similarity of their constituting words; and the similarity between t and R_y^{-t}, which is computed on the basis of the TF-IDF cosine similarity of their texts.

Translation: This baseline (Park et al., 2015) is also a generative model wherein a candidate sentence t of a product P_y generates each review set of all products (expect P_z), which will be used as translations of t. The review sentence t and each of the generated review sets jointly generate t's specifications S_y; and S_y generates the specifications of R_x and the query specifications S_z. Therefore, we have

$$\begin{aligned} score(t, S_y; R, S_z) &\propto p(t, S_y|S_z) \\ &= \frac{p(S_z \mid S_y) \sum_{P_x \in P^{-z}} p(S_x \mid S_y) p(S_y \mid t, R_x) p(R_x \mid t) p(t)}{p(S_z)} \\ &\propto p(S_z|S_y) \sum_{P_x \in P^{-z}} p(S_x|S_y) p(S_y|t, R_x) p(R_x|t) \end{aligned} \tag{20}$$

where P^{-z} refers to the set of all products except for P_z. In this scenario, t is scored on the basis of the similarity between t and R_x; the similarity between P_y, t and R_x; the similarity between products P_z and P_x; and the similarity between products P_z and P_y.

6. Evaluation results and findings

In this section, we compare the performance of our work with those of the baselines on our datasets. To evaluate the baseline models, we separated some products as the test products. These products were regarded as products with no reviews. That is, we excluded the reviews of the test products from the other reviews. These reviews comprised the test set, and the remaining reviews were regarded as the training set. As suggested by Park et al. (2015), we chose the top 50 qualified products by number of reviews as the test products to obtain a statistically reliable gold standard dataset. We highlight that the textual input was preprocessed using NLTK. We removed stopwords, tokenized the words, split the sentences, stemmed the words, and extracted the review sentiments using this toolkit. Given that there are some hyperparameters in our proposed approach, we first discuss how the hyperparameters were tuned.

6.1. Hyperparameter tuning

One of the main parameters of our proposed approach is the number of topics used when embedding the textual attributes through LDA. Given that the size of the topic set can significantly impact the quality of the derived topic model, we select an appropriate number of topics in the trained LDA by using the natural nonparametric generalization of LDA, called the hierarchical Dirichlet process (HDP) mixture (Teh et al., 2005), wherein the number of topics is not required a priori and is learned on the basis of the observed data. We used the microscopes-lda package of Python[4] to deploy HDP. We ran the HDP model for our datasets and selected the best-performing model in terms of its perplexity on 500 iterations.

Now, the next set of parameters included the structured attributes that were selected to represent the structured representation of the products. For this purpose, we chose those attributes that were available across all products in each of the categories in our dataset. More specifically, for the Digital Camera category, we adopted the 'Manufacturer,''Product Type,'and 'Resolution'attributes, and for the MP3 Player category, we used the 'Manufacturer,''Product Type,'and 'Digital Storage'attributes to constitute the structured product representations. These attributes are suggested as the most important attributes in Park et al. (2015) as well. For the movie dataset, we did not use any product specifications, since we found that they did not provide any added value in our work. It is worth noting that we could have also utilized some other product attributes such as 'Average product rating,''Number of reviews,'and 'Number of ratings.'However, the main goal of the embedding phase is to effectively identify the relation between cold and non-cold products because attributes such as 'Number of reviews'and 'Number of ratings'are essentially unreliable for cold products; therefore, including such attributes would negatively impact how cold and non-cold products are related to each other. For instance, two highly similar products (one cold and one non-cold) might be determined to be dissimilar because their number of ratings, reviews, and average rating are not the same, which is undesirable for our problem.

In terms of the hyperparameters of the neural architecture used for learning the product embeddings, we set the parameters on the basis of the recommended default parameter settings reported in Liao et al. (2018). As such, we set the dimension of the embedding vectors to 20, the number of negative samples to 10, and the number of epochs to 20. Furthermore, when building the summarization graph, we employed two threshold values, namely, α and β, to build an unweighted graph. We performed a 10-fold cross-validation strategy to obtain the best values for α and β, separately. The best setting for α was determined to be 0.5 for all datasets. The best values for β were 0.47, 0.5, 0.46, 0.1, 0.3 on Datasets 1 to 5, respectively. Finally, the damping factor, d, introduced in Section 4.1.1 was set to 0.85, as widely recommended in the literature (Kim and Lee, 2002).

6.2. Results

The performances of the various baseline methods on the five datasets introduced in Table 1 in terms of the ROUGE-2 and TF-IDF cosine similarity metrics are reported in Tables 4 to 7. As we noted earlier, ROUGE-2 metrics perform evaluation based on bi-gram matching between the actual reference reviews and the selected sentences, while the TF-IDF Cosine similarity metric considers the weights of the words in computing the similarity between the reference and selected sentences. In other words, ROUGE-2 can be considered to be primarily a syntax-based matching metric, while the TF-IDF cosine similarity metric moves closer to a semantic interpretation of content. On the other hand, the ROUGE-2 metric offers precision, recall, and F1-score, thus providing us with the opportunity to analyze the performance of the

[4] https://datamicroscopes.github.io/hdp.html

Table 4
The Results of the ROUGE-2 Precision Metric. Bold font with † shows statistical significance over the best baseline measured using paired *t*-test with p-value <0.05.

	Dataset 1 (Digital Camera)		Dataset 2 (MP3 Player)		Dataset 3 (Digital Camera-Cold)		Dataset 4 (MP3 Player-Cold)		Dataset 5 (Movie)	
	@100	@200	@100	@200	@100	@200	@100	@200	@100	@200
QL (Ponte and Bruce Croft, 1998)	0.043	0.038	0.053	0.039	0.029	0.026	0.045	0.036	0.004	0.005
MEAD (Radev et al., 2000)	0.058	0.033	0.036	0.029	0.058	0.033	0.040	0.037	0.012	0.007
MEAD-SIM (Park et al., 2015)	0.052	0.047	0.050	0.047	0.059	0.050	0.050	0.047	0.011	0.008
RevSpecGen (Park et al., 2015)	0.033	0.028	0.063	0.055	0.031	0.024	0.054	0.047	0.009	0.008
Translation (Park et al., 2015)	0.040	0.040	0.049	0.038	0.060	0.054	0.082	0.063	0.010	0.008
EbS	**0.261**†	**0.140**†	**0.262**†	**0.256**†	**0.153**†	**0.085**†	**0.098**†	**0.095**†	**0.020**†	**0.015**†
EbS +	0.063	0.039	0.141	0.096	**0.111**†	**0.087**†	**0.222**†	**0.172**†	0.009	0.006
$EbS_{sentiment}$	**0.165**†	**0.089**†	**0.152**†	0.091	0.055	0.069	**0.090**†	**0.093**†	**0.020**†	**0.015**†
EbS $+_{sentiment}$	0.041	0.035	0.147	**0.112**†	**0.106**†	0.075	**0.226**†	**0.196**†	0.004	0.005

different approaches from various aspects. For the estimation of the performance of the summarization systems, a common approach is to evaluate the generated summaries at the maximum length of K words against some reference summaries. Summaries that exceed the size limit will be trimmed down.[5] Since the average length of existing reviews in our datasets is 141, we chose two values for K, i.e., 100 and 200. These values are suggested by the main baseline (Park et al., 2015) as well. Therefore, we report the findings based on reviews with 100 and 200 words, reported as '@100'and '@200'in the results tables. It should be noted that we refer to our method as Embedding-based Summarization (EbS) when using the cosine similarity metric to construct the summarization graph and as EbS + when using the TF-IDF cosine similarity metric. Moreover, $EbS_{sentiment}$ refers to the EbS model that considers sentiment features as additional features for review nodes in the embedding graph, while EbS $+_{sentiment}$ augments the EbS + model with sentiments. We should further mention that the boldface numbers associated with the dagger sign (†) signify statistical significance at a p-value of less than 0.05 based on a paired *t*-test against the best-performing baseline method.

As can be seen in Table 4 and with respect to ROUGE-2 precision, the EbS method significantly outperformed the other baselines on all five datasets, especially on Datasets 1 and 2. To clarify the impact of dataset characteristics on the performance of our proposed method, we analyzed several dataset characteristics reported in Table 1, including the number of products, number of reviews, number of users, average number of RPP, average number of RPU, and average number of SPR. Based on the formulation of the attributed heterogeneous graph, the number of products, reviews, and users directly determines the number of nodes in this graph, while the average number of RPP and the average number of RPU affect the structure of the attributed heterogeneous graph, i.e., they will determine the number of edges in the graph. The higher the RPP and RPU metrics are, the richer the structure of the graph would be. As can be seen in Table 1, Datasets 1 and 2 contain higher values in terms of the number of users, products, and reviews, as well as the RPP and RPU metrics rather than Datasets 3 and 4. This leads to a larger and richer graph in the first phase of our proposed approach, and, therefore, EbS will be able to obtain a stronger structural representation for modeling entities. Furthermore, the larger corpus of reviews is, the larger the text information for LDA would be, which enhances the process of extracting topics for attribute encoding. Considering the values of the RPP, RPU, and SPR metrics for Dataset 5, we can conclude that the structure of the embedding graph is rich for this dataset due to the high values of the RPP and RPU metrics. However, the textual attributes built by the LDA model are not very representative due to the short reviews and low SPR metric. For these reasons, EbS showed better performance on Datasets 1 and 2 than on Datasets 3, 4, and 5 in terms of the ROUGE-2 precision metric. It still, however, managed to outperform the other baselines on Datasets 3, 4, and 5 in the same metric.

The results in Table 4 show that the other version of our proposed approach, EbS +, also outperformed the baselines in terms of the ROUGE-2 precision metric in a statistically significant manner. However, the results of EbS + were lower than those of EbS on this metric in most of the cases. The reason was due to the impact of the TF-IDF weights of words in computing the similarity of two sentences. The simple cosine similarity metric does not include any weights for words and considers the two sentences as a bag of words, while the TF-IDF cosine similarity metric considers the TF-IDF weight of each word. As an example, let us consider the phrase 'lens resolution'in the corpus of the Digital Camera category. Since the frequency of this word in this corpus is high, the TF-IDF weight of this word would be low. As a result, sentences that contain this phrase have a lower chance of being considered as *important sentences* when computing the sentence similarity using the TF-IDF cosine similarity metric than the chance using the simple cosine similarity metric. However, we know that such a phrase should not be penalized in the context of review selection for Digital Cameras as it shows relevance, and, hence, ignoring the importance of this phrase would lead to a lower precision when performing sentence selection. The expectation is that, as the size of the corpus increases, the frequency of such words or phrases increases as well, which would negatively impact the ROUGE-2 precision performance of the model that relies on the TF-IDF cosine similarity metric, i.e., EbS +. It is interesting to note that, as the datasets and the review corpus become smaller, such a negative impact is not observed as the impact of IDF is reduced overall. This can be observed on the smaller datasets including Datasets 3 and 4.

While EbS can retrieve several sentences that are highly relevant to the product, it may also retrieve several sentences that are similar to each other. This will cause a degree of redundancy in the selected sentences by EbS. Therefore, the expectation is that EbS would not be as strong in terms of ROUGE-2 recall compared to its performance on precision given the precision-recall tradeoff. Moreover, for Datasets 3 and 4, which contain less information, as discussed earlier, this issue may naturally be more apparent. The results in Table 5 are consistent with these expectations since we observed that the results of EbS on the ROUGE recall metric for Dataset 3, which has the lowest amount of information, were competitive with the other two main baselines, where the difference was not statistically significant, i.e., translation (0.11 vs. 0.11 for @100 and 0.139 vs. 0.141 for @200) and MEAD-SIM (0.11 vs. 0.12 for @100 and 0.139 vs. 0.150 for @200). Nevertheless, the EbS approach still outperformed the other baseline methods on Datasets 2, 4, 5, and also on Dataset 1 for the @100 evaluation in terms of the ROUGE-2 recall metric. It is worth noting that while Dataset 4 is a sparsely populated dataset, it contains lengthier reviews than those of Dataset 3 (SPR 16.7 vs. 9.6), where EbS was able to obtain superior

[5] https://duc.nist.gov/duc2007/tasks.html

Table 5
The Results of the ROUGE-2 Recall Metric. Bold font with † shows statistical significance over the best baseline measured using paired *t*-test with p-value <0.05.

	Dataset 1 (Digital Camera)		Dataset 2 (MP3 Player)		Dataset 3 (Digital Camera-Cold)		Dataset 4 (MP3 Player-Cold)		Dataset 5 (Movie)	
	@100	@200	@100	@200	@100	@200	@100	@200	@100	@200
QL (Ponte and Bruce Croft, 1998)	0.077	0.099	0.080	0.114	0.055	0.099	0.072	0.105	0.053	0.060
MEAD (Radev et al., 2000)	0.049	0.076	0.081	0.091	0.106	0.160	0.130	0.160	0.085	0.095
MEAD-SIM (Park et al., 2015)	0.111	0.150	0.092	0.110	0.120	0.150	0.091	0.100	0.092	0.116
RevSpecGen (Park et al., 2015)	0.075	0.123	0.100	0.138	0.085	0.123	0.097	0.127	0.065	0.092
Translation (Park et al., 2015)	0.091	0.130	0.108	0.153	0.110	0.141	0.100	0.135	0.072	0.112
EbS	0.111	0.135	**0.111**†	**0.180**†	0.110	0.139	**0.150**†	**0.200**†	**0.127**†	**0.149**†
EbS+	**0.131**†	**0.162**†	**0.173**†	**0.197**†	**0.162**†	**0.216**†	**0.211**†	**0.240**†	0.102	0.133
$EbS_{sentiment}$	0.068	0.120	**0.229**†	**0.235**†	**0.123**†	**0.196**†	**0.157**†	**0.214**†	**0.127**†	**0.147**†
EbS $+_{sentiment}$	0.120	0.130	**0.127**†	0.150	**0.211**†	**0.241**†	**0.233**†	**0.268**†	0.040	0.081

results compared to those of the other methods in terms of the ROUGE-2 recall metric.

An additional interesting observation is the positive impact of adopting the TF-IDF cosine similarity metric for measuring sentence similarities, as was done in EbS+. While EbS assigns high scores to sentences that carry frequent and important words and, as such, selects sentences that contain such words, the EbS+ variant decreases redundancy and obtains higher recall rates owing to the consideration of the TF-IDF values. As such, EbS+ showed statistically significantly better performance than those of the baselines in all five datasets in terms of the ROUGE-2 recall metric. Note that since Dataset 5 contains short reviews, the TF-IDF cosine similarity metric for measuring sentence similarities was not effective to decrease redundancy and increase the recall metric, so it did not perform better than the EbS model in terms of the reall metric. Finally, the results of the ROUGE-2 F1-score are reported in Table 6. The behavior of the comparative approaches in terms of this metric on the five datasets was similar to (consistent with) the behavior observed from the precision metric in Table 4. Like in the ROUGE-2 precision and recall metrics, the EbS and EbS+ methods outperformed the other baselines. It can be further observed that the EbS method shows the best performance when dealing with larger information sets such as Datasets 1, 2, and 5, while EbS+ is most suited for smaller corpora containing mostly cold products, as represented in Datasets 3 and 4.

Regarding the results of the sentiment-enhanced models, i.e., $EbS_{sentiment}$ and EbS $+_{sentiment}$, we can see that equipping the proposed model with the sentiment of the reviews did not improve the performance in most cases. We found that, when sentiments are added as attributes to review nodes, it can lead to undesirable relations between products and reviews primarily because these two entities will be considered to be related to each other if they share the same sentiment. The ideal scenario would be to deduce product and review relations when both review topics and review sentiments are the same.

It should be noted that, in datasets that contains less information, incorporating the sentiments of reviews can improve the efficiency of our methods. As can be seen in our cold datasets, the positive effect of sentiments is observable. For instance, Table 5 shows that, in terms of the ROUGE-2 recall metric, on Dataset 3, the EbS model improved from 0.110 to 0.123 for @100 and from 0.139 to 0.196 for @200 by incorporating review sentiments. Another example can be observed in the same table on Dataset 4, where sentiments offered improvements to the EbS+ model (from 0.211 to 0.233 for @100 and from 0.240 to 0.268 for @200).

We also analyzed the performance of the methods on the basis of the TF-IDF cosine similarity between the selected sentences and the actual review. The obtained results on this metric, as reported in Table 7, showed the superior performance of our proposed methods on the datasets.

In summary, we found that our proposed approaches, namely, EbS and EbS+, are effective in selecting review sentences for a product that does not have reviews yet. We specifically find that.

1. EbS, which does not consider term and document frequency when calculating review similarities, shows better precision on datasets that have a larger number of reviews, while EbS+, which does consider frequency information, performs well on cold datasets with fewer reviews.
2. EbS+ shows overall better performance on the recall metric as it uses term frequency information and leads to the selection of a highly diverse set of review sentences.
3. Overall and in terms of the F1-score metric, both EbS and EbS+ show a statistically significantly better performance over the baselines. However, EbS shows a better performance on datasets that have a larger number of reviews, while the performance of EbS+ is much more significant on cold datasets.
4. Incorporating review sentiments as additional attributes for review nodes does not lead to noticeable improvement to the EbS and EbS + variants and would make improvements in some cold situations

Table 6
The Results of the ROUGE-2 F1-score Metric. Bold font with † shows statistical significance over the best baseline measured using paired *t*-test with p-value <0.05.

	Dataset 1 (Digital Camera)		Dataset 2 (MP3 Player)		Dataset 3 (Digital Camera-Cold)		Dataset 4 (MP3 Player-Cold)		Dataset 5 (Movie)	
	@100	@200	@100	@200	@100	@200	@100	@200	@100	@200
QL (Ponte and Bruce Croft, 1998)	0.055	0.055	0.063	0.057	0.037	0.041	0.055	0.053	0.007	0.009
MEAD (Radev et al., 2000)	0.053	0.046	0.024	0.043	0.053	0.046	0.050	0.051	0.019	0.013
MEAD-SIM (Park et al., 2015)	0.072	0.071	0.064	0.065	0.079	0.075	0.064	0.064	0.020	0.014
RevSpecGen (Park et al., 2015)	0.046	0.046	0.076	0.078	0.046	0.040	0.069	0.068	0.016	0.015
Translation (Park et al., 2015)	0.055	0.061	0.077	0.079	0.067	0.059	0.090	0.085	0.018	0.014
EbS	**0.154**†	**0.137**†	**0.154**†	**0.211**†	**0.127**†	**0.105**†	**0.118**†	**0.127**†	**0.035**†	**0.028**†
EbS+	0.085	0.063	**0.154**†	**0.129**†	**0.131**†	**0.124**†	**0.216**†	**0.200**†	0.017	0.011
$EbS_{sentiment}$	**0.096**†	**0.102**†	**0.182**†	**0.131**†	0.076	**0.102**†	**0.114**†	**0.129**†	**0.035**†	**0.027**†
EbS $+_{sentiment}$	0.061	0.055	**0.091**†	0.078	**0.125**†	**0.110**†	**0.229**†	**0.226**†	0.007	0.009

Table 7
The Results of the TF-IDF Cosine Similarity Metric. Bold font with † shows statistical significance over the best baseline measured using paired *t*-test with p-value <0.05.

	Dataset 1 (Digital Camera)		Dataset 2 (MP3 Player)		Dataset 3 (Digital Camera-Cold)		Dataset 4 (MP3 Player-Cold)		Dataset 5 (Movie)	
	@100	@200	@100	@200	@100	@200	@100	@200	@100	@200
QL (Ponte and Bruce Croft, 1998)	0.214	0.260	0.227	0.230	0.214	0.269	0.200	0.210	0.400	0.380
MEAD (Radev et al., 2000)	0.240	0.200	0.334	0.343	0.230	0.247	0.180	0.211	0.315	0.352
MEAD-SIM (Park et al., 2015)	0.338	0.371	0.253	0.268	0.360	0.400	0.264	0.269	0.335	0.372
RevSpecGen (Park et al., 2015)	0.280	0.300	0.236	0.266	0.300	0.360	0.240	0.260	0.240	0.251
Translation (Park et al., 2015)	0.340	0.380	0.300	0.323	0.350	0.390	0.290	0.300	0.256	0.267
EbS	**0.410**†	**0.430**†	**0.557**†	**0.585**†	**0.401**†	**0.450**†	**0.554**†	**0.561**†	**0.663**†	**0.668**†
EbS+	0.340	0.371	**0.400**†	**0.380**†	**0.374**†	0.385	**0.558**†	**0.562**†	0.371	0.410
$EbS_{sentiment}$	0.351	**0.397**†	**0.390**†	**0.390**†	**0.451**†	**0.459**†	**0.402**†	**0.447**†	**0.659**†	**0.667**†
EbS $+_{sentiment}$	0.311	0.340	**0.357**†	0.341	**0.367**†	0.323	**0.539**†	**0.556**†	0.229	0.262

in terms of the ROUGE-2-recall metric.

The results suggest that, while both EbS and EbS+ provide statistically significant improvement over various baselines on all five datasets, EbS would be the preferred method for datasets with a larger number of reviews, while EbS+ would be better suited for cold datasets. In the end, we would like to point to the generalizability of our approach again. We examined our approach in two main domains, movies and e-commerce. The structured specifications have been expressed in detail and comprehensively in the e-commerce domain, while the movie domain contains fewer specifications for movies. The main comparative approaches rely on the structured specifications of products. We showed that in both domains our approach outperformed the baseline models. Those approaches are not even applicable in some websites such as Amazon, which lack such structured specifications, while the proposed approach is applicable on a wide range of domains and websites.

6.3. Qualitative analysis and discussion

To present a qualitative comparison between our proposed approach and the state-of-the-art baselines, we explored some sample sentences from EbS as well as from the three baselines, namely, translation, MEAD-SIM, and QL approaches. The reason we show translation and MEAD-SIM is that they are stronger than the other baselines from the same family of methods, namely, RevSpecGen and MEAD. Here, we include only the EbS model from among the variations of our work because it performs better on real datasets than the other variations. This section will be complemented by an analysis of the other variations of our work on different datasets in Appendix A. We will discuss how and why the selected sentences are appropriate and where they deviate from the actual reviews. We randomly selected a sample cold product in the Digital Camera category of CNET called 'Olympus E-10 Digital SLR' Digital Camera and show selected review sentences for it in Table 8. For this product, EbS selected a sentence that included the 'Superb image quality'phrase as a part of the first sentence and 'This camera for the price is the camera that ends all cameras'as the third sentence. In the top 50 sentences, EbS included many sentences that discuss the high quality of the camera such as 'Great camera,''Best camera,'and 'This camera is the best camera in this class of cameras.'When manually checking the reviews for this product, we could see that over 80% of the actual reviews for this product explicitly discuss the high quality of this product either generally speaking or when talking about its high-image quality. There existed both types of review sentences in the sentences selected by EbS.

In contrast, when exploring the retrieved review sentences by the translation model, we found sentences such as 'Light weight bad picture quality, very slow shutter, low sensitivity, I bought that from officemax for my trip,''Horrible picture quality! picture quality is the one thing you absolutely don't want to compromise on size, screen size, good battery time and that's about it,'and 'Horrible picture quality, noisy camera, terrible in low light and the low light modes don't help!.' While such review sentences were placed in the top 10 sentences retrieved by the translation model, they do not accurately portray the reviews on this product. There were also similar inaccurate review sentences retrieved by MEAD-SIM such as 'Last camera you will ever buy,''Easy yet can be complex, truly film quality photos,'and 'Limited accessories available.'Similar instances appeared in the QL model such as 'It is very unfair that I should have to pay for the repairs of a faulty product that they sold me'and 'won't turn on; lens doesn't open/close; won't take pictures.'

Let us explore how such review sentences were selected by each approach. The translation model does not consider the relationship between review sentences and products, and primarily focuses on the existence of the same words within the product specification in the review sentences. For instance, in the above examples, the translation model detected image quality as an important attribute; however, it made a mistake in choosing the sentences that truly describe the picture quality of the cold products. This could be because the products in the top 10 sentences of the translation model included some products whose picture quality is not close to that of the product of interest. More precisely, the cold product 'Olympus E-10 Digital SLR'has an overall rating of 8.5 from CNET users, while the translation model retrieved some irrelevant sentences, which were selected from reviews of products that are in the same family but have much lower ratings from the users. For instance, 'Olympus FE-230'and 'Olympus study 710'have similar specifications to the product of interest; however, their overall ratings are 5.7 and 5.6, respectively. Therefore, given that the translation model is a generative model that selects reviews on the basis of the translation of product specifications to reviews, it failed to address such cases. To some extent, MEAD-SIM also suffered from a similar problem as it relied on the similarity between two product specifications when selecting review sentences. The QL method is a softer version of the translation model since it works on the basis of the existence of the product specification words in the review sentences. As a result, the sentences selected by this method contained specification words such as 'picture,''lens,''SLR,'and 'camera'; however, those sentences are often far from the actual reviews.

Now, while the translation and MEAD-SIM approaches can fail when similar products but with different qualities exist, our proposed approach might face challenges when many highly similar review sentences are observed in different yet related products. In such cases, while the precision of EbS is quite high, it will resort to retrieving similar sentences and, hence, it performs poorly on recall. For instance, for the 'Canon Powershot SD630'product, EbS retrieved very similar review sentences in the top 10 sentences such as 'Great camera,''What a camera!!!,''This camera has it all,''This camera does it all!,''This camera does it all for me.'Although these sentences matched the actual review patterns from the users, they negatively impacted the recall metric, as reported in Table 5. However, in contrast, the translation model is

Table 8
Sample Selected Sentences by EbS, translation, MEAD-SIM, and QL Approaches for the 'Olympus E-10 Digital SLR'Digital Camera on Dataset 1.

EbS	MEAD-SIM (Park et al., 2015)
*Superb image quality, good tonal response and dynamic range good high iso performance, built-in image stabilization, generally fast and responsive in use twist and swivel screen useful for certain shooting types jpeg engine, make contrast detect AF pretty slow in live view moderate LCD screen resolution slightly lower absolute resolution than rest of class conclusion *JPEG engine makes the most of the sensor's output Useful in-camera RAW processing option (though lacks any preview, which is limiting). *This camera, for the price, is the camera that ends all cameras.	*Last camera you will ever buy. *Easy yet can be complex, truly film quality photos. *Limited accessories available…yet being a life long advanced amateur photographer (40 + years), I am all but certain this will be the last camera that I will buy. *It delivers truly high quality film equivalent photo in the "auto" modes, I will reiterate plural "modes" a child can operate it very successful. *This camera also allows the user to do as they wish with available manual modes, try doing that with other digitals.
Translation (Park et al., 2015)	**QL (Ponte and Bruce Croft, 1998)**
*I bought it for under water usage and liked the ruggedness. *Light weight bad picture quality, very slow shutter, low sensitvity I bought that from officemax for my trip. *Shock proof, water proof, big range of built-in shooting settings (all accessible via a simple dial) rugged. * Competitive price, good lenses, broad feature set none to date I'm not what you would call an early adopter of new technologies. *Horrible picture quality! Picture quality is the one thing you absolutely don't want to compromise on size, screen size, good battery time and that's about it… *Horrible picture quality, noisy camera, terrible in low light(and the low light modes don't help.	*The seller - digitalrev - has a reputation for being incredibly unsupportive in accepting warranty claims, and it is very unfair that I should have to pay for the repairs of a faulty product that they sold me. *Inform yourself: http://www.fixya.com/…http://answers.yahoo.com/…http://www.digitalcamera-hq.com/digital-cameras/…all the complaints are the same: won't turn on; lens doesn't open/close; won't take pictures *I researched this camera on cnet and www.digitaladvisor.com before purchasing it. *See my review http://www.epinions.com/… *The best slr camera of any type I ever owned!!

capable of selecting attribute-specific sentences since it scores sentences on the basis of the occurrence of specification words in the sentences. For instance, it retrieved sentences that are focused on detailed attributes such as 'internet radio,''app store,''internet browser,'and 'widescreen viewing'for the 'Apple iPod 5G'product; however, in many cases, the retrieved sentences for each of these attributes were not quite accurate.

Unlike the translation and EbS approaches, the MEAD-SIM approach has a main drawback in that it selects exactly the same sentences for products that are highly similar to each other. This is because the only factor in the MEAD-SIM model that is affected by the products is product-product similarity. For example, in the MP3 Player category, MEAD-SIM selected the same review sentences for all of the following products: 'Apple iPod Nano (1 GB),''Apple iPod Nano (second generation 8 GB),''Apple iPod Nano (fifth generation 80 GB),''Apple iPod Shuffle,''Apple iPod Classic,'and 'Apple iPod Touch.'This is despite the fact that each of these products has received different reviews from the users as they each have a different set of strengths and weaknesses.

Overall, the translation model favors specific reviews that mention product attributes but could fail to identify the correct review sentences for these aspects, and, as such, it is quite specific yet not quite accurate. MEAD-SIM returned similar reviews for similar products; therefore, its specificity and accuracy depend on the neighboring products. Our approach is quite accurate in retrieving correct review sentences, but can, at times, favor more generic review sentences. Fig. 6 visually compares these three methods.

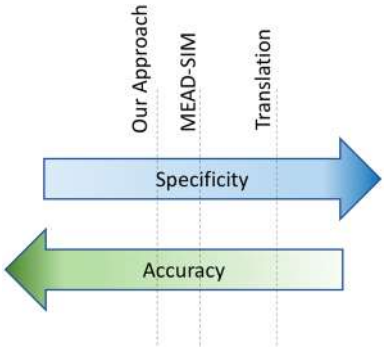

Fig. 6. An Intuitive Comparison between the EbS, MEAD-SIM and translation Models.

7. Concluding remarks

In this paper, we have proposed an approach for the retrieval of review sentences for cold products by selecting relevant sentences from other products by simultaneously considering user, product, and review interactions. We have proposed a generalizable framework for representing the attributes of and the relationships between various entities of an e-commerce website such as users, products, and reviews through an attributed heterogeneous graph. On the basis of this graph representation, we have discussed how the nodes can be embedded into a dense low-dimensional representation that makes user, product, and review nodes comparable. The comparable representations of products and reviews are then used to select reviews for a cold product of interest. We have empirically evaluated our proposed approach against strong state-of-the-art baselines and analyzed both quantitatively and qualitatively the pros and cons of each approach.

This study opens up exciting directions for future works. (1) It would be interesting to extend our work by enriching the selected sentences for cold products by exploring exogenous information sources such as content shared on microblogging services such as Twitter. We intend to investigate a review transfer approach to retrieve those tweets that express user feedback about cold products. The main challenge of this would be related to the disparities between users' feedback and the informal language they use in expressing their opinions about products. Our proposed generalizable approach in this paper can potentially incorporate information from social networks, which can help identify and selectively choose tweets that can be used to generate reviews for cold products. Examples of such information include users' social contextual features and relations with other users, users' demographic information, textual content of users in the network, and relations between different entities in the networks, e.g., users, hashtags, and trends. This information can be incorporated into our proposed approach, to enrich either the structure of the embedding graph or the attributes of its nodes. (2) Another direction of our future work would be to apply our proposed framework to other languages. Our proposed approach does not consider deep linguistic features. The only language-dependent step is the preprocessing step where we perform tasks such as tokenizing, stemming, and sentence splitting. Therefore, it would be rather straightforward to apply our approach to other languages. (3) We are also interested to examine our work by incorporating the exact similarity degrees to construct the summarization graph. That is, instead

of an unweighted graph, it may be helpful to use a weighted version of the summarization graph by using the similarity degrees produced by the embedding graph as well as the degrees of similarity between sentences (see Section 4.2). This graph is much larger than its unweighted counterpart and would require much more computational resources to process, but, at the same time, might produce improved performance. (4) Finally, another area that we would like to explore is the capture of additional qualifiers when preprocessing review sentences. For instance, it would be ideal to distinguish between the two cases of a review that mentions "This camera *does not* (vs does) allow the user to use manual modes." We will explore in a future work the use of our work on capturing qualifiers (Cruz et al., 2016) such as *negation* for processing product reviews.

Acknowledgments

The third author kindly acknowledges funding from the Natural Sciences and Engineering Research Council of Canada (NSERC). We would also like to thank the anonymous reviewers for their constructive feedback that helped improve this work.

Appendix A. More sample sentences

In Section 6.3, we qualitatively analyzed the retrieved sentences from Dataset 1 by using the EbS approach as well as the baseline methods for a random cold product. Here, we report more sentences for other variations of our work as well as for the two strongest baselines, i.e., translation and MEAD-SIM.

Table A.9 complements Table 8 and contains the output of the EbS+, $EbS_{sentiment}$, and EbS $+_{sentiment}$ models on Dataset 1 for the same product, namely, 'Olympus E-10 Digital SLR'Digital Camera. Tables A.10, A.11 represent the retrieved sentences of shows the results for the ment, EbS+, $EbS_{sentiment}$, and EbS $+_{sentiment}$ models on Dataset 3 (Camera Cold) for the same product.

Tables A.12 and A.13 show the retrieved sentences obtained by the translation, MEAD-SIM, EbS, EbS+, $EbS_{sentiment}$, and EbS $+_{sentiment}$ models for the 'Dell Digital Jukebox DJ (20 GB)'MP3 player on Dataset 2, and Tables A.14 and A.15 report reviews on the same product but on Dataset 4. Table

Table A.9
Sample Selected Sentences by EbS+, $EbS_{sentiment}$, and EbS $+_{sentiment}$ Approaches for the 'Olympus E-10 Digital SLR'Digital Camera on Dataset 1.

EbS+	$EbS_{sentiment}$	EbS $+_{sentiment}$
Great camera! First DSLR.	What a camera!	
Camera build feels solid.	This camera does it all for me.	It seems that so many of these parts need to be replaced that I am too far down the list for service within the 6 + weeks the camera has been at the service center.
Low light, fast, very sharp.	I had to with all my other cameras.	
We bought this camera as a step up from all the point and shoot cameras we have had over the years. That lens is phenomenal.	Is it the camera?	After the LCD cracked without being bumped or dropped (even the clear plastic protective covering didn't crack), I have spent the past three months dealing with service centers and paid an additional $150 for parts and labor, and the camera is still not fixed.
If you don't require top notch video you have cheaper alternatives.	This is that camera!	
I prefer to have my camera and video camera in 1 device rather than carry 2 separate devices.	This camera has it all.	Don't let the low price fool you small and easy to use; good quality pix for bottom end camera extremely slow, low battery life, poor service while my recently purchased s50 worked, I was more or less satisfied.
Silent lens, zoom abilities, etc.	If this is to by your only camera.	
Here are our reasons why we selected the gh1.	This camera does it all.	
Price has come down since the initial cnet review but it's still pricey.	As they were on the camera.	For the seemingly low price, the camera performed OK (not great) and took good quality pix.
Finally, cnet was a great resource to read about all these cameras.	This camera does it all.	
	This is the camera for you!	
	This camera, for the price, is the camera that ends all cameras.	
	There are better cameras out there.	
	Do not buy this camera.	
	The old camera isn't perfect by any stretch (its slow shutter makes it difficult to take sports pictures), but for the most part using it involved three simple steps…	

Table A.10
Sample Selected Sentences by EbS, translation, and MEAD-SIM Approaches for the 'Olympus E-10 Digital SLR'Digital Camera on Dataset 3.

EbS	Translation (Park et al., 2015)	MEAD-SIM (Park et al., 2015)
Would not recommend this camera.	Shock proof, water proof.	Biased evaluation the size, the stabilization, the advanced menus, the color resolution the pentaprism. I think that the evaluator doesn't understand the camera and the differences between it and the competition, this evaluation is quite biased, I have used olympus cameras and the overall image quality beats the competition, I personally don't like video on an SLR, but this is a matter of taste, for the menus I think that they are quite good, when you use the camera you see that you can access normally used settings quite fast and don't have to dig into innumerable menus to get what you want, unfortunately this review is not a good guide and just guides potential customers to nikon and canon, and don't even mention sony or pentax, quite bad …
This is not a camera.	Big range of built-in shooting settings (all accessible via a simple dial) rugged.	
As they were on the camera.	Superb IQ, fast start-up, manual focus, decent built-in flash.	
This camera is what it is.	Easy-to-navigate interface no built-in view finder this guy (or gal, if you prefer) is a good size for an ICL, it straddles the boundary between fits-in-a-large-pocket and need-an-extra-bag, depending on which lens you use.	
Its a camera?	Incredibly sharp elecronic finder.	
This camera does it all.	Underwater use, price, small size, ease of use, tough!	
This camera is the best camera I have ever owned and I am addicted to cameras!	I compared it with the current crop of 4–5 megapixel cameras from kodak, nikon, fuji and olympus, all retailing at $149–199.	
This camera is great!!	Flash will not remain turned off even with reset turned off! very good picture file.	
All cameras have their pros and cons, but if you know how to use a camera and have patience, this is the camera for you.		
Great camera.		
This is a great camera.		
Great camera!		
All in all a good camera.		
Has everything except picture quality looks like an slr		
Easy to operate built-in 30x optical zoom lens manual, aperture and shutter control settings menu is fairly easy to navigate comfortable to hold affordable		

Table A.11
Sample Selected Sentences by EbS+, $EbS_{sentiment}$, and EbS $+_{sentiment}$ Approaches for the 'Olympus E-10 Digital SLR'Digital Camera on Dataset 3.

EbS+	$EbS_{sentiment}$	EbS $+_{sentiment}$
I love this camera.	Has everything except picture quality looks like an SLR	
This camera is awesome! love the touch screen.	Easy to operate	All in all a good camera.
I love it.	Built-in 30x optical zoom lens	Easy to use.
I am in love with this camera.	Manual, aperture and shutter control settings	Small, compact, easy-to-use.
Not a bad camera after all.	Menu is fairly easy to navigate, comfortable to hold, affordable compared to cameras with similar features sub-par picture quality,focus is inconsistent, slow response time, noisy video, LCD hard to see in daylight, image sizes are listed as L, M or S rather than megapixels or dimensions if you only want to view your photos on your computer and at a size no larger than your monitor, then this is a great camera.	Good value for price.
Easy to use.		The wide angle mode helps with this too.
It takes amazing pictures!		Turn out well.
Overall, it is a very nice camera.		I would turn the camera on, take a picture or two, and turn it off again. Can't turn off in auto mode.
Good camera for a beginner, very featureful.		Fast! awesome camera!
Do not buy this camera.		Great quality I was in love and for the price?
All in all a good camera.		I don't know what happened with the other buyers.
Great pictures quality.	Large LCD screen, simple turn-on-and-take, connection and upload process, portability and great looking camera, picture quality with flash is sublime!	Perfect camera great value!
There are better cameras out there.		It will take a perfect great must have pocketable hi-mp camera powerful power shooter, tons of control, easy usage…
Very, very, disappointed! I am so disappointed		Best chunk of aluminum I ever had the fortune to buy
Best chunk of aluminum I ever had the fortune to buy.		
Easy to use.		
Good for the price, easy to use.		
Very good at its price.		

A.16 shows the results for the EbS, translation, and MEAD-SIM models for the 'Ai Weiwei: Never Sorry' movie on Dataset 5.

As we have noted earlier and seen in Table A.9, unlike EbS, the EbS+ model avoids the redundancy of some frequent words in the selected sentences by considering the TF-IDF weights of words when computing sentence similarities. This causes a higher recall while reducing precision. Here, we observed this fact in the results returned by the EbS+ model, where sentences expressing the overall evaluation (e.g., 'Great camera!') of the product appeared much less frequently than in the results returned by the EbS model. On the other hand, some other important words managed to appear in the selected sentences. For instance, aspects such as 'light,' 'lens,' 'build,' and 'speed,' which are aspects that are more or less emphasized in the actual reviews, were included in the retrieved sentences by EbS+, while EbS did not retrieve some of them.

We can see some irrelevant sentences in the sentiment-based variants, i.e., $EbS_{sentiment}$ and EbS $+_{sentiment}$, such as 'Is this the camera?,''As they were on the camera,' or sentences with opposite directions such as 'Don't buy this camera,'and 'The old camera isn't perfect.' This could be explained by the negative effect of considering the sentiments of reviews on the relations between products and reviews, which has been discussed in Section 6.2. A clearer example of such incorrect relations could be seen in the sentences retrieved by EbS $+_{sentiment}$, where, for the product 'Olympus E-10 Digital SLR' with an overall score of 8.5, some review sentences from product 'Pentax Optio S50' with an overall score 6.6 were selected. It is obvious that the relations between these products were not represented well in the EbS $+_{sentiment}$ model.

Table A.10, A.11 show the top selected sentences from Dataset 3 (Camera Cold) for the same product 'Olympus E-10 Digital SLR.' Table A.10 illustrates the results of the EbS, translation, and MEAD-SIM models. This product has received positive reviews from the reviewers; however, the retrieval of sentences, such as 'Would not recommend this camera,' as the first sentence shows that the retrieved sentence was not very accurate. The reason could be due to the sparsity of Dataset 3. However, overall, the majority of the retrieved sentences by the EbS model still resembled the actual reviews where several sentences represent the overall general perception of the product and some other sentences include more specific aspects such as 'ease of use' and 'lens quality.'

Moreover, we observed that the retrieved sentences by the translation method contained concepts such as 'shock proof, water proof,' 'fast start-up,' 'built-in flash,' and 'viewfinder.' While these sentences refer to some important features of the camera, they do not resemble the actual reviews.

Table A.12
Sample Selected Sentences by EbS, translation, and MEAD-SIM Approaches for the 'Dell Digital Jukebox DJ (20 GB)'MP3 player on Dataset 2.

EbS	Translation (Park et al., 2015)	MEAD-SIM (Park et al., 2015)
		It is solid mp3 player, well built.
This is not an mp3 player. music player:)	I had a lot to learn about using the tiny thing, and let me admit, I was daunted at first.	No current software updates.
This is my ideal "mp3 player" - get it!	However, the eq is quite responsive and allows for some serious adjustment.	Great for any portable music-ing needs so long as you're using vista or older OS and have deubox installed.
This is what an mp3 player should be all about and I stress mp3 player.	Large screen, gps, expandable data storage.	Bone conduction sound transfer works well.
What a player! Those other players cannot do this.	Was able to get a sansa view with product replacement plan (highly recommend getting one for mp3 players!).	Music is easy to add as it comes up as just another drive on your computer.
Do other players do this?	Before I get to my complaints and/or issues, let me say what I did like.	This model doesn't come with its own goggles.
If so, this is the player for you	This player's sound is really great, and the option to change visuals with one-click is nice, along with the feature to delete a song off the device while your playing it if you don't like it anymore.	My Reeboks are actually a good goggle but with the Swimp3 tucked in there water can seep in.
Its a player which has all you want of mp3 player	First off I'd like to talk about the issue raised by cnet about bass response.	You're forced to tighten the goggle to where the nose piece presses into the flesh.
Great, great player.	I was opposed to getting an ipod, because I am a pc user.	You can't expect a fancy interface from a small, sleek mp3 player in the pool.
The sound quality is great.		So you have to know ahead of time where the controls are on your head and what they do.
Great sound quality.		
Great! again, great sound quality.		
I can't prove that like the other facts since I don't have the equipment but I don't think they can lie about this!!		
Long battery life!		
Great battery life.		
But I know it's not for games it's for music, so it's perfect for what it's intended for.		

In the MEAD-SIM sentences, sentences were selected from 'Olympus E-620.' While this product is highly similar to the product of interest, less relevant sentences from 'Olympus E-620' were selected by the MEAD-SIM approach. In Table A.11 for Dataset 3, we can see that the TF-IDF cosine similarity in the EbS+ model enhanced the quality of the selected sentences; for instance, the overall evaluation of the product mentioning 'good price,' 'picture quality,' and 'ease of use,' was selected correctly in the retrieved sentences. However, there were also some incorrect retrievals such as 'Very, very, disappointed.' We can see that the negative effect of incorporating sentiments decreased here since only a few sentences such as 'Large LCD' in the $EbS_{sentiment}$ and 'Best chunk of aluminum I ever had the fortune to buy' in EbS $+_{sentiment}$ seemed to be selected incorrectly. The reason for this has been elaborated in Section 6.2.

Tables A.12,A.13,A.14,A.15 show the retrieved sentences by the baselines for a random cold product: 'Dell Digital Jukebox DJ (20 GB)' MP3 player. Tables A.12 and A.13 show the selected sentences from Dataset 2, and Tables A.14 and A.15 show the retrieved sentences from Dataset 4. The overall rating for this product is 6.3, as assigned by CNET users. We can see from Table A.12 that the EbS approach contains review sentences referring to the good or great overall quality and functionality of the product. The EbS approach also detected that 'sound quality,' as well as 'battery life,' is another feature that seems to be satisfying. Surprisingly, all of these aspects are mentioned in a considerable number of actual reviews with the same polarity in sentiments. The translation approach identified and retrieved a negative review sentence for this product as the first sentence, which is not aligned with the overall perception of the customers for this product. In the following retrieved sentences, we can see that the translation model identified aspects such as 'EQ,' 'large screen,' 'GPS,' and 'expandable data storage.' Most of these features are important aspects; however, many of the sentences that contain these phrases are semantically irrelevant to the product because the translation model failed in building semantic connections between the products and the sentences that contain those aspects. The MEAD-SIM approach suffered from the same drawback since irrelevant phrases such as 'software update,' 'Bone conduction sound transfer,' and 'SwiMP3' were seen in the top selected sentences by this method. The other problem of MEAD-SIM, which is also mentioned in Section 6.3, is its bias on product specification similarity. For example, for Dataset 2, the selected sentences by the MEAD-SIM approach for the products 'Creative Zen Micro' with a total rating of 6.9, 'Creative Zen 8G' with a total rating of 6, and 'Creative Zen Vision: M' with a total rating of 8 are exactly the same solely because they have the same specifications.

Table A.13 shows the selected sentences by EbS+, $EbS_{sentiment}$, and EbS $+_{sentiment}$ on Dataset 2 for the same product. We can see that, when using the EbS+ model, the redundancy of the sentences such as 'I love this player' was reduced and that some other features managed to be included in the top sentences. For instance, in addition to 'good sound quality,' which was also detected in the EbS model, 'good price,' 'no FM tuner,' and 'supported music formats' were considered by the EbS+ model in the top sentences. Here, we can also see some irrelevant features such as 'image quality' and 'supporting SD and SDHC' by the EbS+ model.

We can see that, while the sentiment-enhanced models, i.e., $EbS_{sentiment}$ and EbS $+_{sentiment}$, reduced the importance of generic aspects, they included a wider range of relevant aspects such as 'nice design' and 'ease of use.' However, they also included other irrelevant aspects as well, such as 'poor fonts,' 'poor buttons,' and 'high price.'

Table A.14 lists the sentences selected by EbS, translation, and MEAD-SIM for the same product from Dataset 4. We can see that, although the few top sentences retrieved by the EbS model covered the main features, including the overall quality, sound, size, price, and ease of use, the model also had some errors in selecting sentences, such as 'Video quality is not too good' and 'Wait to buy until audio comes out with the software to let this player use DRM protected music,' since the features in these sentences are not observable in the actual reviews.

We can see that the translation model also identified and covered many features, such as 'weight,' 'sound,' 'size,' 'price,' 'customer support,' 'ease of use,' 'screen,' 'memory,' 'battery,' 'looking,' 'USB plug,' 'playlist,' 'buttons,' 'DRM files,' and 'Lexaras tech support' in the few top sentences; however, not all of them are consistent with the actual reviews of the mentioned product. Specifically, sentences mentioning 'weight,' 'sound,' 'size,' 'price,' 'ease of use,' 'screen,' 'memory,' 'battery,' 'looking,' and 'USB plug' are consistent with actual reviews, while sentences containing 'no playlist,' 'buttons,' 'DRM files,' 'Lexaras tech support,' and 'customer support' are inconsistent with the real reviews.

Table A.13
Sample Selected Sentences by EbS+, $EbS_{sentiment}$, and EbS $+_{sentiment}$ Approaches for the 'Dell Digital Jukebox DJ (20 GB)'MP3 player on Dataset 2.

EbS+	$EbS_{sentiment}$	EbS $+_{sentiment}$
It can't record video and FM radio.	This mp3 player is the best sounding mp3 player I have heard.	This mp3 player is the best sounding mp3 player I have heard.
If you love music, you will love this!	Sound is great.	Sound is great.
I love it!	Great mp3 player.	It wasn't that great.
I love this music player.	It would be a great player if they updated a few things.	It would be a great player if they updated a few things.
If you love music, you will love this!	This is my ideal "mp3 player" mp3 player- get it!	I just wanted to say something about this.
I love this player.	Good battery life.	It's just ok. nice screen, ok sound quality.
The sound quality is excellent.	This is the player!!!!	But still, I didn't like how it sorted files through folders.
Good sound quality.	For an mp3 player, this has to be the best.	I'm using my friend's computer right now and saw this. But I wouldn't recommend this since I see a lot of mp3s that are cheaper and are a lot better.
Overall, for the price this is an excellent buy.	It's good, nice design and easy to use has good resolution for videos, good sound quality the software is awful!!	The buttons they put on this make it look even more cheaper than it already is.
Very good sound quality.	Easy to use, great!	I like that it was really cheap and that it had nice extras but I watched the review and I had to agree.
Overall I like love this mp3 player.	It wasn't that great.	The font makes it look cheap.
Awesome image. I love it!	The font makes it look cheap.	
I was amazed at how much I love it.	It's just ok. nice screen, ok sound quality.	
love this player.	I'm using my friend's computer right now and saw this. But I wouldn't recommend this since I see a lot of mp3s that are cheaper and are a lot better.	
Excellent sound and image.		
Cannot delete files in music/ movie mode.		
Supports SD cards and SDHC.		
Virtually all music formats supported.		

Table A.14
Sample Selected Sentences by EbS, translation, and MEAD-SIM Approaches for the 'Dell Digital Jukebox DJ (20 GB)'MP3 player on Dataset 4.

EbS	Translation (Park et al., 2015)	MEAD-SIM (Park et al., 2015)
Great little player for the money, I love its size and portability. Excellent player fulfilled my expectations great sound. You can't look at pictures and listen to music at the same time. I really got this player for the space, not for the color screen, but I was pleasantly surprised to like it a lot better than the regular zen micro (which my sister has). Video quality is not too good but still it's a great player. The iPod shuffle is easy to use which makes this a good mp3 player for working. I was looking for a player I could use at work and in the car, but this one wasn't it. This is a great mp3 player; I have been using one for a month now. Wait to buy until audio comes out with the software to let this player use DRM protected music	This players plays mp3 as it suppose too. Good light weight rubber construction. It was 4.99. I love it, for 4.99, but it sounds only OK, and is quite small, but 4.99! Thanks for giving me an errand for a present, bro!) Works fine. Great for kids. No playlist, ridiculous customer support, can't resume track from where you left off. Great form factor, simple to use, backlit screen Buttons did not work after 1 day of use, does not play DRM files I am absolutely shocked at the ineptitude of Lexar's tech support. Reasonable price, good memory capacity, and it's cute! Light, direct USB plug, can use rechargeable batteries to maximize play time, nice looking, …	It is solid mp3 player, well built. No current software updates. Great for any portable music-ing needs so long as you're using vista or older OS and have deubox installed. Bone conduction sound transfer works well. Music is easy to add as it comes up as just another drive on your computer. This model doesn't come with its own goggles. My Reeboks are actually a good goggle but with the Swimp3 tucked in there water can seep in. You're forced to tighten the goggle to where the nose piece presses into the flesh. You can't expect a fancy interface from a small, sleek mp3 player in the pool. So you have to know ahead of time where the controls are on your head and what they do.

The MEAD-SIM bias on product specification similarity is more obvious in Table A.14. Both Datasets 2 and 4 contain a product named 'Dell DJ 20' whose specifications are similar to those of the cold product 'Dell Digital Jukebox DJ (20 GB).' We can see from Tables A.12 and A.14 that MEAD-SIM selected the top sentences for the cold product from the reviews of product 'Dell DJ 20' for both datasets. Although Dataset 2 is much richer than Dataset 4 and contains more sentences and products, we can see that the MEAD-SIM approach relied on the reviews of that single product and ignored other sentences. Regarding the actual reviews of these two products, we can see that, despite their similarity in specifications, they differ much in their reviews. For example, features such as 'software update,' 'Bone conduction sound transfer,' and 'SwiMP3' are completely different from the actual reviews of 'Dell Digital Jukebox DJ (20 GB).'

Table A.15 shows the sentences selected by EbS+, $EbS_{sentiment}$, and EbS $+_{sentiment}$ for a cold product. Here, we can see that more accurate features were included in the EbS+ model sentences than those in EbS. However, still, some sentences were selected incorrectly, such as sentences that point to video quality.

The analysis of the movie reviews differs from commercial products since the movie domain consists of a wider range of 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. While an e-commerce product can be described by its features and descriptions to a great extent, there are several implicit factors in each movie that affect the users' opinion which are not captured in the structured specifications of the movie. In the movie domain, the specifications may include genre, director, writer, and year. It has been shown that in the movie domain relying entirely on the product attributes to address the cold product problem may cause several problems (Zhu et al., 2019). There are several examples of movies with similar features in which users have different opinions about them. It is noted that we also have such examples in the e-commerce domain, but this issue is more apparent in the movie domain. Moreover, We found the diversity of the feature values in our movie dataset is high, so, there are very few films with common attributes such as director and writer. As a result, applying feature-based approaches, e.g., translation and MEAD-SIM, might not be the best choices for these domains.

Table A.15
Sample Selected Sentences by EbS+, $EbS_{sentiment}$, and EbS $+_{sentiment}$ Approaches for the 'Dell Digital Jukebox DJ (20 GB)'MP3 player on Dataset 4.

EbS+	$EbS_{sentiment}$	EbS $+_{sentiment}$
Great little player for the money, I love its size and portability. Love the unit and the long battery life great sound. Easy to use, battery life and sound quality are great. Video quality is not so good. The video on the device is not great. Really good video quality. Easy to use, battery life and sound quality are great. Video quality is not too good but still it's a great player. That's why we buy these things for good sound quality. Great sound quality. Great video screen. Easy to use, great quality, small. Again, great sound quality. I tried FM recording feature.. Pretty good record quality. I can now record songs that I like while I am listening to them.	But if we are going to let ourselves be robbed…this is a great little unit for the price. Video quality is not too good but still it's a great player. Excellent player fulfilled my expectations. Great sound, straightfoward and easy to use interface and touchpad. You can't look at pictures and listen to music at the same time I really got this player for the space, not for the color screen, but I was pleasantly surprised to like it a lot better than the regular zen micro (which my sister has). The iPod shuffle is easy to use which makes this a good mp3 player for working. I was looking for a player I could use at work and in the car, but this one wasn't it.	Great for audio books and other long audio files. Love the unit and the long battery life great sound. Easy to use, battery life and sound quality are great. Really good video quality. The video on the device is not great. It can't record video and fm radio. Great video screen. Easy to use, battery life and sound quality are great. Sounds good and so easy to use. Good iPod. That's why we buy these things for good sound quality. Again, great sound quality. The sound quality is good. Great sound quality. Great video screen. Video quality is not so good. Video looks good too, but it is a little small.

Table A.16
Sample Selected Sentences by EbS, translation, and MEAD-SIM Approaches for the 'Ai Weiwei: Never Sorry'Movie on Dataset 5.

EbS	Translation (Park et al., 2015)	MEAD-SIM (Park et al., 2015)
It's a must-see. It's not subtle, but it's effective. It's shocking, it's hilarious but most importantly, it's eye-opening. This one's for them. It's not deep, but then, it's about vanity. What's not to like? But she ends up making a virtue of the actor's zen calm…he's so present, it's as though he's burned into the screen. It's not quite hagiography, but it's almost hard not to get caught up in the film's admiring tone. It's about life. It's a powerful film. He's right on target in his handling of vandyke's story and the film is a bull's eye. I can't prove that like the other facts since I don't have the equipment but I don't think they can lie about this!!	And its muckraking spirit, an anomaly in the age of giving in, is inspiring. It has about four good ideas for a feature documentary, though none of them quite lands. Now into their eighties, the taviani brothers show with this remarkable, fresh and moving drama-documentary they have lost none of that mix of observational rigour and sympathy for the underdog that marked early films like padre padrone. Filtering writer jon savage's non-fiction doorstopper into a mere 78 min …makes this feel like a taster for a more in-depth tv series - but one well worth seeing, nonetheless. Although it's more suited for the small screen, it is a worthy entry nonetheless.	Given the subject, this documentary is stunningly boring Not quite enough actual titan the film's images give a backbone to the company and provide an emotional edge to its ultimate demis That the e-graveyard holds as many good ideas as bad is the cold comfort that chin's film serves up with style and empath Happily for Mr. Chin – though unhappily for his subjects – the invisible hand of the marketplace wrote a script that no human screenwriter could have hoped to matc Even though it's common knowledge that park and his founding partner, yong kang, lost kozmo in the end, you can't help but get caught up in the thrill of the company's astonishing growt

Table A.17
Comparison of the overall functionality for retrieving review sentences between the different methods.

Method	Overall Evaluation
EbS	has an accurate sentence selection but suffers from sentence redundancy in warm datasets, focuses on more important product features, and, hence, has higher redundancy on such features.
EbS +	has fewer redundant sentences, covers a wider range of product features, and is more useful for cold datasets.
Sentiment-enhanced methods	are useful for cold datasets and have shown to be less effective on warm products.
Translation	covers a range of features in the retrieved sentences but is not as accurate as EbS.
MEAD-SIM	is biased toward the similarity of product specifications and ignores the semantic relationship between products.

Table A.16 shows the selected sentences by the EbS, translation, and MEAD-SIM approaches for the movie 'Ai Weiwei: never sorry'. This movie has an overall rating of 8.12 from Rotten Tomatoes users. Here are some actual reviews from the critics for this film: 'A powerful film that teaches us as much about ourselves as it does it's subject, Ai Weiwei: Never Sorry, is a sure bet to be nominated for an Oscar come January 2013.,' 'A sobering, cautionary tale.,'and 'One of the most engagingly powerful movies of the year'. Looking at the other reviews, it is obvious that the overall perception about this film is positive. We can see in Table A.16 that EbS was able to extract correct sentences with this point of view such as 'It's a must-see'and 'It's a powerful film'. This is while the other baselines do not have such correct selections. It is worth mentioning that the EbS + model was able to detect a feature of this film that was mentioned by some critics. Consider this sentence in the actual reviews: 'Top-drawer documentary about the art'. We can observe sentences like this in the first thirty sentences selected by the EbS + model as 'that said, all art deserves biography, and great art deserves recognition'. This shows that the EbS + model is able to extract more specific features than the EbS model. We preferred to not include an extra table with sentences of the EbS + and sentiment enhanced versions of our approach, primarily because there were no more significant observations in the top few sentences. However, The mentioned example suffices to illustrate our point.

We have summarized our observations made from the qualitative analysis in Table A.17.

References

Ali, Syed Muhammad, Noorian, Zeinab, Bagheri, Ebrahim, Ding, Chen, Ding, Chen, Al-Obeidat, Feras, 2018. Topic and sentiment aware microblog summarization for twitter. J. Intell. Inf. Syst. 1–28.

Barjasteh, I., Forsati, R., Ross, D., Esfahanian, A.H., Radha, H., 2016. Cold-start recommendation with provable guarantees: a decoupled approach. IEEE Trans. Knowl. Data Eng. (TKDE) 28 (6), 1462–1474.

Berger, Adam, Lafferty, John, 1999. 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, pp. 222–229.

Cannon, Lynn Elliot, 1969. A cellular computer to implement the Kalman filter algorithm (Ph.D. thesis). Montana State University-Bozeman, College of Engineering.

Coppersmith, Don, Winograd, Shmuel, 1990. Matrix multiplication via arithmetic progressions. J. Symbolic Comput. 9 (3), 251–280.

Cruz, Noa P, Taboada, Maite, Mitkov, Ruslan, 2016. A machine-learning approach to negation and speculation detection for sentiment analysis. J. Assoc. Inf. Sci. Technol. 67 (9), 2118–2136.

Dunning, Ted, 1993. Accurate methods for the statistics of surprise and coincidence. Comput. Linguistics 19 (1), 61–74.

Erkan, Günes, Radev, Dragomir R, 2004. Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. 22 (1), 457–479.

Feng, Yue, Bagheri, Ebrahim, Ensan, Faezeh, Jovanovic, Jelena, 2017. The state of the art in semantic relatedness: a framework for comparison. Knowl. Eng. Rev. 32 (1), 1–30.

Geijn, Robert A., Watts, Jerrell, 1997. SUMMA: scalable universal matrix multiplication algorithm. Concurrency: Practice Experience 9 (4), 255–274.

Gong, Yihong, Liu, Xin, 2001. Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 19–25.

Kim, Sung Jin, Lee, Sang Ho, 2002. An improved computation of the pagerank algorithm. In: European Conference on Information Retrieval, Berlin, Heidelberg, pp. 73–85.

Kingma, Diederik, Ba, Jimmy, 2015. Adam: a method for stochastic optimization. In: International Conference on Learning Representations, pp. 1–15.

Liao, Lizi, He, Xiangnan, Zhang, Hanwang, Chua, Tat-Seng, 2018. Attributed social network embedding. IEEE Trans. Knowl. Data Eng. 30 (12), 2257–2270.

Lin, Chin-Yew, 2004. Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pp. 74–81.

Liu, Mengwen, Fang, Yi, Choulos, Alexander G., Park, Dae Hoon, Hu, Xiaohua, 2017. Product review summarization through question retrieval and diversification. Inf. Retrieval J. 20 (6), 575–605.

Louis, Annie, Nenkova, Ani, 2009. Automatically evaluating content selection in summarization without human models. In: Association for Computational Linguistics 1. pp. 306–314.

Louis, Annie, Nenkova, Ani, 2013. Automatically assessing machine summary content without a gold standard. Computat. Linguistics 39 (2), 267–300.

Luo, Wenjuan, Zhuang, Fuzhen, Zhao, Weizhong, He, Qing, Shi, Zhongzhi, 2015. QPLSA: utilizing quad-tuples for aspect identification and rating. Inf. Process. Manage. 51 (1), 25–41.
Mihalcea, Ra.da., Tarau, Paul, 2004,. Textrank: bringing order into text. In: Proceedings of the 2004 conference on Empirical Methods in Natural Language Processing (EMNLP 2004), pp. 404–411.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: In Advances in neural information processing systems (NIPS 2013), pp. 3111–3119.
Moghaddam, Samaneh, Ester, Martin, 2013. 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, pp. 909–918.
Moratanch, N., Chitrakala, S., 2017. A survey on extractive text summarization. In: Computer, Communication and Signal Processing (ICCCSP), 2017 International Conference on. IEEE, pp. 1–6.
Musat, C.-C., Liang, Yizhong, Faltings, Boi, 2013. Recommendation using textual opinions. In: IJCAI International Joint Conference on Artificial Intelligence, pp. 2684–2690.
Na, Liu, Ming-xia, Li, Ying, Lu, Xiao-jun, Tang, Hai-wen, Wang, Peng, Xiao, 2014. Mixture of topic model for multi-document summarization. In: The 26th Chinese Control and Decision Conference (2014 CCDC). IEEE, pp. 5168–5172.
Nenkova, Ani, McKeown, Kathleen, 2012. A survey of text summarization techniques. In: Mining text data. Springer, pp. 43–76.
Ouyang, You, Li, Wenjie, Li, Sujian, Qin, Lu., 2011. Applying regression models to query-focused multi-document summarization. Inf. Process. Manage. 47 (2), 227–237.
Owczarzak, Karolina, Conroy, John M., Dang, Hoa Trang, Nenkova, Ani, 2012. An assessment of the accuracy oNenkovaf automatic evaluation in summarization. In: Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization. Association for Computational Linguistics, pp. 1–9.
Oya, Masanori, 2015. Centrality Measures of Sentences in an English-Japanese Parallel Corpus. In: 2015, 58.
Park, Dae Hoon, Kim, Hyun Duk, Zhai, ChengXiang, Guo, Lifan, 2015. 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, pp. 393–402.
Ponte, Jay M., Bruce Croft, W., 1998. 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, pp. 275–281.
Poria, Soujanya, Chaturvedi, Iti, Cambria, Erik, Bisio, Federica, 2016. Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis. In: Neural Networks (IJCNN), 2016 International Joint Conference on. IEEE, pp. 4465–4473.
Pourgholamali, Fatemeh, 2016. Mining Information for the Cold-Item Problem. In: Proceedings of the 10th ACM Conference on Recommender Systems. RecSys '16. New York, NY, USA: ACM, pp. 451–454.
Pourgholamali, Fatemeh, Kahani, Mohsen, Bagheri, Ebrahim, Noorian, Zeinab, 2017. Embedding unstructured side information in product recommendation. Electron. Commer. Res. Appl. 25 (1), 70–85.
Qiu, Lin, Gao, Sheng, Lyu, Qinjie, Guo, Jun, Gallinari, Patrick, 2018. A novel non-Gaussian embedding based model for recommender systems. Neurocomputing 278, 144–152.
Radev, Dragomir R., Jing, Hongyan, Budzikowska, Malgorzata, 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In: Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization-Volume 4. Association for Computational Linguistics, pp. 21–30.
Radev, Dragomir R, Jing, Hongyan, Styś, Malgorzata, Tam, Daniel, 2004. Centroid-based summarization of multiple documents. Inf. Process. Manage. 40 (6), 919–938.
Salton, Gerard, Buckley, Christopher, 1988. Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24 (5), 513–523.
Saveski, M., Amin, M., 2014,. Item cold-start recommendations: learning local collective embeddings. In: Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14. ACM, pp. 89–96.
Shi, C., Hu, B., Zhao, X., Yu, P., 2018. Heterogeneous information network embedding for recommendation. IEEE Trans. Knowl. Data Eng. Early Accsess 1-1.
Sun, Yizhou, Han, Jiawei, Yan, Xifeng, Yu, Philip S., Wu, Tianyi, 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In: Proceedings of the VLDB Endowment 4.11, pp. 992–1003.
Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., Wang, X., 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In: International Joint Conference on Artificial Intelligence, IJCAI'15. Springer, pp. 1333–1339.
https://www.accenture.com/t20160624T012737_w_/gr-en/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Strategy_7/Accenture-How-Retailers-Can-Drive-Profit-and-Competitiveness.pdf.
Tan, Jiwei, Wan, Xiaojun, Xiao, Jianguo, 2017. Abstractive document summarization with a graph-based attentional neural model. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1171–1181.
Teh, Yee W,. Jordan, Michael I., Beal, Matthew J., Blei, David M., 2005. Sharing clusters among related groups: Hierarchical Dirichlet processes. In: Proceedings of Advances in neural information processing systems, pp. 1385–1392.
Tripathy, Abinash, Anand, Abhishek, Rath, Santanu Kumar, 2017. Document-level sentiment classification using hybrid machine learning approach. Knowl. Inf. Syst. 53 (3), 805–831.
Ulrich, Jan, Murray, Gabriel, Carenini, Giuseppe, 2008. A publicly available annotated corpus for supervised email summarization. In: Proc. of aaai email-2008 workshop, chicago, USAs.
Wang, Dingding, Zhu, Shenghuo, Li, Tao, Gong, Yihong, 2009. Multi-document summarization using sentencebased topic models. In: Proceedings of the ACL-IJCNLP 2009 conference short papers. Association for Computational Linguistics, pp. 297–300.
Wong, Kam-Fai, Mingli, Wu., Li, Wenjie, 2008. Extractive summarization using supervised and semi-supervised learning. In: Association for Computational Linguistics 1. pp. 985–992.
Xiao, Ding, Ji, Yugang, Li, Yitong, Zhuang, Fuzhen, Shi, Chuan, 2018. Coupled matrix factorization and topic modeling for aspect mining. Inf. Process. Manage. 54 (6), 861–873.
Xiong, Shufeng, Ji, Donghong, 2016. Query-focused multi-document summarization using hypergraph-based ranking. Inf. Process. Manage. 52 (4), 670–681.
Yang, Yinfei, Chen, Cen, Qiu, Minghui, Bao, Forrest, 2017. 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, pp. 675–680.
Zhang, Weishi, Ding, Guiguang, Chen, Li, Li, Chunping, Zhang, Chengbo, 2013. Generating virtual ratings from chinese reviews to augment online recommendations. ACM Trans. Intell. Syst. Technol. 4, 1.
Zhang, Fuzheng, Jing Yuan, Nicholas, Lian, Defu, Xie, Xing, Ma, Wei-Ying, 2016. Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp. 353–362.
Zhang, Yongfeng, Ai, Qingyao, Chen, Xu, Bruce Croft, W., 2017. 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, pp. 1449–1458.
Zhang, Menghao, Binbin, Hu., Shi, Chuan, Bin, Wu., Wang, Bai, 2018. Matrix factorization meets social network embedding for rating prediction. In: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data. Springer, pp. 121–129.
Zhao, W.X., Li, S., He, Y., Chang, E.Y., Wen, J.R., Li, X., 2016. Connecting social media to E-commerce: cold-start product recommendation using microblogging information. IEEE Trans. Knowl. Data Eng. (TKDE) 28 (5), 1147–1159.
Zheng, Lei, Noroozi, Vahid, Yu, Philip S., 2017. Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, pp. 425–434.
Zhou, Liang, Ticrea, Miruna, Hovy, Eduard, 2004. Multi-Document Biography Summarization. In: Proceedings of EMNLP 2004. Barcelona, Spain: Association for Computational Linguistics.
Zhu, Yu, Lin, Jinghao, He, Shibi, Wang, Beidou, Guan, Ziyu, Liu, Haifeng, Cai, Deng, 2019. Addressing the item cold-start problem by attribute-driven active learning. In: IEEE Transactions on Knowledge and Data Engineering.