PAPER

A New Sentiment Case-Based Recommender

Mashaël ALDAYEL^{†,††a)}, Student Member and Mourad YKHLEF^{††}, Nonmember

SUMMARY Recommender systems have attracted attention in both the academic and the business areas. They aim to give users more intelligent methods for navigating and identifying complex information spaces, especially in e-commerce domain. However, these systems still have to overcome certain limitations that reduce their performance, such as overspecialization of recommendations, cold-start, and difficulties when items with unequal probability distribution exist. A novel approach addresses the above issues through a case-based recommendation methodology which is a form of content-based recommendation that is well suited to many product recommendation domains, owing to the clear organization of users' needs and preferences. Unfortunately, the experience-based roots of case-based reasoning are not clearly reflected in case-based recommenders. In other words, the concept that product cases, which are usually fixed feature-based tuples, are experiential is not adopted well in case-based recommenders. To solve this problem as well as the recommenders' rating sparsity issue, one can use product reviews which are generated from users' experience with the product as a basis of product information. Our approach adapts the use of sentiment scores along with feature similarity throughout the recommendation unlike traditional case-based recommender systems, which tend to depend entirely on pure similarity-based approaches. This paper models product cases with the products' features and sentiment scores at the feature level and product level. Thus, combining user experience and similarity measures improves the recommender performance and gives users more flexibility to choose whether they prefer products more similar to their query or better qualified products. We present the results using different evaluation methods for different case structures, different numbers of similar cases retrieved and multilevel sentiment-approaches. The recommender performance was highly improved with the use of feature-level sentiment approach, which recommends product cases that are similar to the query but favored for customers.

key words: recommenders, case-based reasoning (CBR), case-based recommenders, opinion mining

1. Introduction

Recommender Systems (RSs) have been receiving much attention, especially at e-commerce domains. Such systems help users to access complex information spaces by gathering information and pro-actively adapting it according to their interests. One of the main advantages of RSs is the capability to customize the needs and preferences of the individual. They combine ideas from information retrieval

and filtering, user modeling, and human-computer interaction. The largely used recommendation techniques tend to recommend based on the assumption that users have a steady behavior over time and that similar users share similar tastes. Most of these techniques show inadequate performance when new users or infrequent items appear (cold-start problem) whereas others may tend to overspecialize the resulting items. Furthermore, as they tend to ignore the fundamental structure of users' first choice, and do not evaluate the hidden concepts under which items are selected, their accuracy drops when the item has unequal probability distributions [1]–[3].

A better approach, which addresses the above problems, is through a case-based recommendation methodology. The development of such a method is motivated by Case Based Reasoning (CBR), which has played a significant role in its progress. Concepts from CBR can be found in case-based recommenders, which depend on the similarity between item (product) queries and a set of product cases (case base).

However, the connection between case-based recommenders and the experience-based roots of CBR tends to be blurred. They both use similarity assessment as a foundation for query-product comparison, although the concept that product cases, which are usually fixed feature-based tuples, are experiential is not adopted well in case-based recommenders. To solve this problem as well as the recommenders' rating sparsity issue, one can use product reviews which are generated from users' experience with the product as a basis of product information [4], [5].

The increasing popularity of social e-commerce sites has motivated users to write reviews about their items' usages. Furthermore, on-line product reviews have a significant influence on the purchasing behavior of users. Recent research [6] has shown that 88% of consumers are influenced by on line reviews in purchase decision-making. Many companies consider these reviews as a critical source of product feedback and predict sales according to these on-line user reviews. These reviews are often written in the form of textual comment that clarify why the users like or dislike an item, depending on their usage experiences. The system can then take the multi-sided nature of the customers' opinions from their reviews and therefore construct a preference model for products. This cannot be done with recommendation approaches that use overall ratings [7].

In this paper, a personalized case-based recommendation method based on customer review is proposed for use

Manuscript received October 29, 2016.

Manuscript revised February 27, 2017.

Manuscript publicized April 5, 2017.

[†]The author is with Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia.

^{††}The authors are with Information System Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia.

a) E-mail: maldayel@KSU.EDU.SA

DOI: 10.1587/transinf.2016EDP7441

in e-commerce domains. This approach produces recommendation based on a combination of feature similarity and customer opinions. We implemented a case-based recommender using jCOLIBRI, which is a Java framework for building CBR. We evaluate its performance using different methods, various case models, and multi-level sentiments' approaches. The recommender performance was highly improved with the use of the feature-level sentiment approach that recommends product cases similar to the query but favorable to customers.

2. Background

The objective of Knowledge Management (KM) is to leverage the information resources of organizations to support a purposeful activity for achieving a positive definable outcome. One of the main challenges in KM is how to identify the right knowledge at the right time. With the expansion of the Internet and the advances in network technology, a huge amount of on-line resources is growing at an exponential rate. This increases the difficulty and complexity for people to retrieve the information of interest. People need a lot of considerable time and effort to get the target information from the Internet. The progress in information filtering to present the most relevant results to the user can be derived using a recommendation system whereas the construction of KM systems is ideally suited to CBR owing to the close match between their activities [8]. The combination of case-based reasoning and recommender system has been used so successfully in KM systems. The next sections describe this combination in details.

2.1 Recommender Systems

Recommender systems aim to give users more intelligent methods for navigating and identifying complex information spaces especially at e-commerce domains. One of the main properties of recommendation technologies is the capability to customize the needs and preferences of the individual. Recommendation techniques have two main classes: collaborative filtering and content-based approaches; the former depends on user ratings information whereas the latter depends on item descriptions. The collaborative filtering approach recommends items that do not exist in the active user's profile but those that other users have rated highly. In contrast, the content-based approach makes recommendations based on the item's similarity to previous items liked by the target user, without directly relying on the preferences of other users [1], [2]. The collaborative filtering approach recognizes users whose preferences are similar to those of a particular user and recommends items they have liked whereas the content-based approach recommends items similar to those a particular user has liked in the past [9].

Although broadly used in ecommerce, collaborative filtering recommenders still have to overcome scalability and cold-start problems that limit their performance. They can-

not recommend items not yet rated by any user. As a result, they suffer from the first-rater problem which requires such a system to wait until the new item is rated by numerous users to be able to recommend it. They become ineffective when the rating sparsity problem arises because of the reduced coverage of the recommendation space or if users have difficulty in expressing their favorites as scalar rankings on items. Additionally, content-based RSs still suffer from the recommendations' limited diversity and overspecialization problems, which limit the items recommended to users only to similar items that were previously rated. Thus, users cannot find something unexpected [3], [7], [9].

2.2 Case-Based Recommenders

Case-based recommender systems have their origins from CBR methodology, which is one of the most successful machine learning approaches that take advantage of the knowledge-rich representation of the application domain. Mainly, CBR is a problem-solving methodology that is closely related to natural human reasoning in everyday situations when facing new problems. CBR uses previous experiences to solve new problems based on the CBR hypothesis that is "similar problems have similar solutions". Past cases (situations that have been practiced and learned before) are stored in the case base, which contains cases of previously solved problems and their solutions. New problems are solved by transmitting and adapting solutions that were used for similar problems in the past. CBR retrieves past practices for reuse in the target problems. Such process requires revising previous-case solutions before applying them. Then, CBR retains the positive problem-solving experiences for further reuse. All of these processes make up the traditional 4R processes of CBR: retrieve, reuse, revise, and retain [2], [10], [11]. This knowledge may be reused when required without applying the entire procedure from scratch, or when highlighting a procedure that should be eliminated in a similar problem. Therefore, CBR is able to expand the problem-solving performance over time [3].

CBR has played a significant role in the progress of an important type of recommender system known as case-based RS. Case-based recommendation is a form of content-based recommendation that is well suited to many product recommendation domains. It relies on a structured representation of cases, typically as sets of well-defined characteristics with their values. For example, items or products are represented using a well-defined set of features (e.g., shape, price, color, etc.) to make fine-grained judgments and high-quality suggestions concerning the similarities between items and users' queries. Therefore, this approach has proven to be very successful in many e-commerce sites, owing to the clear organization of users' needs and preferences (Fig. 1) [1], [2]. The main benefit of using CBR in product recommendation is the capability to recommend alternatives that may be suitable if the system cannot find a solution that accurately matches a user's query [12].

Since the basic idea behind CBR is that similar prob-

Fig. 1 Illustration of the role of the case-based recommender system

lems have similar solutions, the similarity measurement used for finding similar cases is a fundamental concept of CBR, and is an important factor of its successful application. Case-based RSs follow the general CBR cycle and depend on the fundamental CBR concepts of similarity and retrieval [3]. Furthermore, there are noticeable matches between the retrieval in CBR and the handling of the user query in RSs; the user query works as a new problem specification, the item descriptions are cases in the case base, and the recommended items are retrieved based on their similarity to the user's request. Hence, case-based RSs can be distinguished from other types of content-based systems using two mechanisms: (1) item representation as a case, i.e., case model, and (2) recommendations based on retrieving cases similar to a user's query, i.e., similarity assessment [1], [2], [10].

2.2.1 Case Model

Cases are modeled using two components; problem description and proposed solution. The product features specified by the user (customers' demands) are shown in the problem component whereas the product itself is shown in the solution component of the case. Let P, S be the subsets of the problem descriptions and solutions respectively. Then a case can be defined as ordered pairs $c = (p, s)$, where $p \in P$, $s \in S$, and C is the case base in a CBR represented by a set of recorded cases $C = (P, S)$. Then, CBR retrieves cases with similar descriptions and adapts their solutions according to the requirements of the target problem [3], [13].

In product recommendation, the system's task is to find a product description that suits the demand of a customer as much as possible. Even if the exact product is not available or does not exist, the most similar available products will be recommended. Moreover, the traditional view of CBR is not reflected in these product recommenders as the stored cases do not represent problem-solution components of the past cases. They are usually just product descriptions. The RS matches a problem (user query) directly with possible solutions (products). This works fine for product recommendation because here problems and solutions are defined by identical terms. Some researchers considered this approach as utility-oriented matching since the similarity measure is simply used to estimate the utility of recognized solutions

(product descriptions) for a specified problem [13].

Owing to the lack of a clear division between the problem part and the solution part of a product case, there are two approaches of case-based models used in e-commerce domains [3], [14]. The first model considers cases as solutions represented only by the descriptions of the available products [15]. More specifically, the solution is the case itself (product recommended) and the problem is the customer's query, which is basically a partial description of the preferred product. Although this model is used by many case-based RSs, the case does not store the user preferences or the context that initiates the recommendation problem [14].

The second model maintains user preferences by forming cases with the customers' queries (the problem part) and the preferred products (the solution part). The recommender takes a user's query and constructs the recommended list in two steps. First, it forms the retrieval set that includes products most similar to the user's query as well as the most selected products in the past cases that are similar to the current case. Second, majority voting strategy is used for the retrieval set because the same product may be referred by different cases in the retrieval set [14].

Case model is one of the vital issue in CBR for selecting the features of a case and choosing the representation structure that is better suited to a particular knowledge domain. Therefore, the case model's task is associated with (1) the choice of related features, (2) the design of the indexes, and (3) the knowledge construction in a specific case. The goal of indexing is to speed up the search process targeting the most relevant contexts. This is accomplished through the design of additional data structures that detect the case features used in the case similarity measurement. Moreover, indexes can accelerate the retrieval procedure to get quick access to the cases to be compared with the user's case problem. For example, in a medical diagnosis system, characteristics such as profession, gender, or age may be less important than the characteristics describing the symptoms [11].

2.2.2 Similarity Assessment

Because case-based recommenders depend on structured case representations, they take advantage of structured methods in similarity assessment. Therefore, they avoid the problem of retrieval inflexibility (stonewalling) that is found in content-based recommenders. Similarity-based retrieval is a valuable feature in case-based recommenders, giving benefits over traditional exact matching and constraint practices. If a user's query is over-specified (i.e., no product precisely matches the query), similarity-based retrieval techniques can retrieve a set of useful similar products. In the same way, if the user's query is underspecified (i.e., too many items exactly match the query), similarity techniques permit ranking products and even eliminating those with the lowest similarity marks [1], [2].

More complex similarity metrics can be used in case-based recommenders relying on an explicit mapping of case features and the availability of specific feature level similar-

Table 1 Global similarity measures at the case level.

Global Similarity	Description
Weighted Block-City (Manhattan)	Calculates the average of the local similarities of its weighted features. It is the Minkowski distance with $r = 1$
Euclidean	It is the Minkowski distance with $r = 2$
Minkowski	Calculates using the following equation: $[\sum_{i=1}^n (t_i - c_i)^r]^{1/r}$
Maximum	Returns the maximum value among the local similarities as the global similarity value

Table 2 Local similarity measures at the feature level.

Feature	Local Similarity	Description
Nominal	Equal	$\begin{cases} 1 & \text{if } t_i = c_i \\ 0 & \text{otherwise} \end{cases}$
Numeric	Symmetric (similar distance whether higher or lower)	$1 - \frac{ t_i - c_i }{(Max_i - Min_i)}$
	INRECA More is better (asymmetric)	$\begin{cases} 1 & \text{if } t_i < c_i \\ 1 - \frac{(t_i - c_i)}{t_i} & \text{otherwise} \end{cases}$
	INRECA Less is better (asymmetric)	$\begin{cases} 1 & \text{if } t_i > c_i \\ 1 - \frac{(Max_i - c_i)}{(Max_i - t_i)} & \text{otherwise} \end{cases}$

ity knowledge. These global metrics measure similarity at the case level, i.e., between cases, as shown in Table 1.

The common method is to use a weighted sum metric to measure the similarity between a user's query case, t , and a candidate item case, c , such as that shown in Eq. (1).

$$Similarity(t, c) = \frac{\sum_{i=1}^n w_i \times Sim(t_i, c_i)}{\sum_{i=1}^n w_i} \quad (1)$$

In Eq. (1), $Sim(t_i, c_i)$ calculates the local similarity at the feature level between two cases concerning the i -th feature and w_i is the i -th feature's weight. Each weight encodes the relative importance of a particular feature. Hence, the products most similar to the user's query can be recommended. Furthermore, the feature level similarity metrics can be calculated using two different methods: symmetric and asymmetric. Symmetric similarity metric is used if there is no bias in favor of either the upper or the lower value of a numeric feature; otherwise, asymmetric will be used. Moreover, evaluating the similarity of nominal features necessitates extra domain knowledge [1], [7]. Examples of local feature-based similarity measures are presented in Table 2 where Max_i and Min_i refer to the maximum and the minimum values of the i -th feature.

In e-commerce domains, product recommendation could permit users to specify feature weights (w_i) to define the significance of certain product features in their purchasing decision [15]. The recommendation session starts with the user query which encodes the customer preferences and ends either when the customer chooses a product or leaves the session without any product choice. The former is considered as a successful recommendation case whereas the

latter is considered as a failed one [14].

Pure similarity-based retrieval approaches tend to lack diversity in recommendations and, thus, limit user choices. If the recommended items are very similar to each other, this may decrease the possibility of recommendation success. Users prefer to receive a beneficial set of item alternatives if the first recommendation is inappropriate. This increases the opportunity that at least one of the recommended items will satisfy the user. Therefore, many alternative approaches [16] have been proposed in to improve recommendation diversity while, at the same time, preserving item similarity. To sum up, similarity assessment is obviously a significant concern for CBR and case-based RSs, which necessitates the modeling of similarity knowledge, i.e., feature-based similarity metrics and weighting functions [1], [2].

3. Customer-Oriented Review Recommendation

On-line product reviews have a major influence on the purchasing behavior of consumers. Recent research [6] has shown that 90% of consumers read these reviews and 88% of them trust them as much as personal recommendations. Many companies consider these reviews as a valuable source of product feedback and predict sales based on their basis. Customer reviews are a type of explicit feedback wherein users are requested to evaluate items, specifying the degree of relevance or interest to users. Customer reviews about a single item are collected and offered to the users to simplify the decision-making process. For instance, a customer's feedback at Amazon.com or eBay.com can help users in determining whether an item has been valued by the community [9].

Data acquired from reviews are used to enhance RSs whether the data are sparse or not by providing further information about user preferences [7]. Case-based RSs usually represent cases (products) based on the offered features in product catalogs. Nevertheless, these product features barely capture the experiences of the users of the products. Instead, it focuses on the technical properties and physical description of a product. To solve this problem, one can use product reviews which are generated from customers' experience with the products as a basis of product information in case-based recommendation [4], [5]. The purpose of using customer reviews is to enrich the informative product case base on the basis of feature opinions to optimize the matching of user's preferences. Moreover, it will increase the quality and accuracy of product ranking. Thus, products with higher sentiment values can be recommended to the user [7].

The process of customer-review recommendation is not straightforward; it goes through different steps until it can be utilized in case-based recommenders. Products or items are recommended based on the reviews from selected customers who have been long-term users of those products. This selection process is done by the provider of the recommendations e.g., Amazon.com where customers are considered as experts as they are interested in the products, have

Fig. 2 Workflow of the customer-review process in recommenders

used them a number of times, and know the characteristics of these products. Based on their experience with a product, the customers will give their assessments which are taken as a review. They can also rate the product positively or negatively. Given only the experience of customers about a product, a feedback is collected from all of them including their opinion of whether to use the product or not. Moreover, an expert's personal profile is not established. RSs collect the reviews of experts and give them to other customers searching for recommendations using the following process:

1. Customers use a RS for a purchase. RS treats the customers as experts if they often (more than a specified number of times) come to the RS and buy products.
2. After the customers buy the products, the RS gives them a feedback form to obtain their opinion about the products' value to customers.
3. From the feedback, ratings are calculated and the feedback expert's name and the description are also gathered. However, the personal profile of the expert is not stored.
4. When new customers use RS and look for products, the RS gives them a list of feedback about the product. Hence, the customers can pick their product of interest.

Figure 2 shows the workflow of the customer-oriented review and recommendation process [17].

Although textual customer reviews are helpful, they can overload active users because they have to read and interpret each review to determine whether it is positive or negative, and to what degree. Advanced machine learning techniques can be utilized to enable case-based recommenders to automatically perform this type of sentiment analysis. This analysis includes using natural language processing (NLP), text analysis, and computational linguistics to identify and extract the feature information in customers' reviews.

The overall approach of case-based recommenders of products (P) based on customers' reviews is presented in Fig. 3. There are four basic steps: (1) finding the beneficial product features, (2) connecting each feature with a sentiment score based on the content of user reviews, (3) generating product cases by combining the features and the sentiment scores, and (4) retrieving recommended cases given a target query [4], [5]. Feature extraction and sentiment calcu-

Fig. 3 Architecture of the case-based recommender for a reviewed product

lation based on the user reviews are outside the scope of this paper. We focus more on how to model experience cases using sentiment scores and improve recommendation methods accordingly. In case-based RSs, a user's preference is shown as a query of the product that he/she likes [7]. A case for a product is represented with a set of product features and their sentiment scores. These scores are mined from a set of reviews of a specific product. The sentiment of each feature is calculated based on all reviews containing this feature. The system then takes each product case as a set of features including the sentiment scores and retrieves those that are similar to the user's query case. Hence, the products that are most qualified and similar to the user's query can be recommended.

3.1 Experience-Based Case

To model experiential cases, i.e., cases that includes sentiment scores, we describe each product (P) by a set of features $F(P) = \{F_1, F_2, F_3, \dots, F_n\}$ extracted from the product's meta-data. Each feature F_i is linked to a set of positive, negative, or neutral sentiment labels which are used to calculate the feature-level sentiment score $Sent(F_i, P)$ as shown in Eq. (2) [4]; $Pos(F_i, P)$, $Neg(F_i, P)$, and $Neut(F_i, P)$ stand for the number of positive, negative, and neutral sentiment labels for feature F_i . The overall sentiment of all product features $Sent(P)$ is calculated as the average of the sentiment scores of each feature, as illustrated in Eq. (3).

$$Sent(F_i, P) = \frac{Pos(F_i, P) - Neg(F_i, P)}{Pos(F_i, P) + Neg(F_i, P) + Neut(F_i, P)} \quad (2)$$

$$Sent(P) = \frac{\sum_{i=1}^n Sent(F_i, P)}{n} \quad (3)$$

Then, the sentiment scores of the product features can be used in two ways: (1) at the feature level $Sent(F_i, P)$ and (2) at the product level $Sent(P)$ which is calculated as the average of the overall sentiment score for all product features. Consequently, the product case, i.e., $Case(P)$ can be specified by Eq. (4) and Eq. (5).

$$Case(P) = \{[F_i, Sent(F_i, P)] : F_i \in F(P)\} \quad (4)$$

$$Case(P) = \{[F_i, Sent(P)] : F_i \in F(P)\} \quad (5)$$

In other words, we combine the product features with their sentiment scores in case-based recommenders and model the case structure to represent the experiments at user reviews.

3.2 Recommendation Assessment

Unlike traditional case-based recommenders, which tend to depend entirely on pure similarity-based assessment in ranking products with respect to user query, our approach adapts the use of the sentiment scores along with feature similarity throughout recommendation. We propose two methods to evaluate the ranking between a candidate product case, c , and a user's query, t . The first method considers the product case representation at the feature-level sentiment $Sent(F_i, P)$, which is used as a dynamic weight in similarity based assessments (as shown in Sect. 2). The method can be modeled in Eq. (6), where $sim(t_i, c_i)$ is a traditional local feature similarity metric that has a value between $[0, 1]$ (defined earlier in Eq. (1)), and $Sent^*(F_i, P)$ is the feature-level sentiment normalized to $[0, 1]$.

$$Rank(t, c) = \frac{\sum_{i=1}^n Sent^*(F_i, P) \times sim(t_i, c_i)}{\sum_{i=1}^n Sent^*(F_i, P)} \quad (6)$$

The second method considers the product case representation using the overall product-level sentiment $Sent(P)$, which is used to adjust the ranking using a weighted combination of similarity and sentiment as formulated in Eq. (7), where $Sent^*(P)$ is the product-level sentiment normalized to $[0, 1]$ and w is the weight controlled by a user [5].

$$Rank(t, c) = (1-w) \times Similarity(t, c) + w \times Sent^*(P) \quad (7)$$

This gives users more flexibility to choose whether they prefer products more similar to their query or better qualified products. Product level and feature level sentiment-based approaches improve the performance of case-based RSs, as shown in our experiment (Sect. 5).

4. Related Works

On-line product reviews have a significant influence on the purchasing behavior of users [6]. Recently, semantic analysis and probabilistic topic models from NLP have been successfully used in information retrieval and then in RSs. The main idea is to emphasize the criteria of the products in customer reviews by analyzing topics, which are sets of words from a given vocabulary. Then, documents are generated with probability distributions over topics. These methods result in better performance in accuracy measurement and in dealing with sparsity problems. They totally depend on the rating concept which can be implicitly or explicitly used as the key input to RS [3], [17]. RecoProd [18] is a content-based recommender that utilizes sentiment techniques. It uses product-level sentiment scores and the costs of the products to recommend products to the customers.

Furthermore, the RS was developed using CBR to overcome problems such as information overspecialization and

cold-start. Case-based recommendation approaches, including conversational and critiquing-based recommenders, are limited in their usage of unified case descriptions centered on technical features. This type of case representation is not appropriate in any experiential context [4], [5]. On contrary, cases generated from reviews are experiential. They can be produced from the product features that customers consider in their reviews, i.e., features linked to the opinions of these customers. To cope with the increased importance of user reviews, researchers [4], [5] have discovered new approaches in case-based recommenders to structure product case representation by taking advantage of the opinion data from users' reviews. They developed similarity-based ranking depending on the product features' sentiment. The retrieved product case to be matched with the users' query is made using both feature sentiment and feature popularity (i.e., frequency of feature occurrence in the product reviews). Sentiment information is uncommon in a recommendation context even though it provides a better opportunity to match between products based on a feature-by-feature sentiment comparison. We used similar concepts with different equations in this paper without using cosine similarity.

Another research [3] proposed a case-based recommendation approach for market basket analysis using a hierarchical model. However, there is still little research in the field of case-based recommenders. Features extraction and sentiment calculation based on user reviews are outside the scope of this paper. The aim of this study was to use this information in product case representation that can be used for recommendation rather than for review filtering or classification. Moreover, this study demonstrates the applicability of the case-based recommender approach based on customer reviews in e-commerce domain using different similarity measurements. We recommend product cases that are similar to the query but favorable to customers.

5. Experiments and Result Analysis

We implemented a case-based recommender using jCOLIBRI, which is a Java framework for building CBR [19]. Our experiment was designed for the objective of comparing the evaluation of case-based RSs using different data set representations and multilevel-sentiment ranking calculation methods. The best approach uses human subjects, but owing to the lack of sufficient subjects to execute the experiment, we instead used an alternative approach, i.e., the leave-one-out cross-validation (LOOCV) approach, to simulate the human-computer dialog process. LOOCV is a special case of J-fold cross-validation where the number of users J is set to 1. LOOCV takes each case in the case base, removes it from the case base, and uses it as a query for that execution cycle. We used different evaluation measurements relevant to CBR and information retrievals (IRs) to solve the problem illustrated in Fig. 1, which explains the role of case-based recommender: given a new query that has some feature values, the output will be a list of related past

Table 3 Case structure as a CBR case model

Feature Name	Local Similarity	Possible Values
<i>Case Description Features</i>		
Model, Body, Color, Doors	(Equal)	Text (String)
Miles, Price	(Less is better)	Number
Year, Speed	(More is better)	Number
CarCode1		ID number
<i>Case Solution</i>		
CarCode2		ID number
Manufacturer		Text (String)

Table 4 Case structure as a product recommender case model

Feature Name	Local Similarity	Possible Values
<i>Case Description Features</i>		
Model, Body, Color, Doors	(Equal)	Text (String)
Miles, Price	(Less is better)	Number
Year, Speed	(More is better)	Number
Manufacturer		Text (String)

items depending on the used ranking approach.

The next sections describe how to model an e-commerce data set with sentiment scores: case structure and query structure. Afterward, a detailed retrieval and ranking of the case-based recommender are presented based on different sentiment approaches: at the feature level and at the product level. Next, we discuss the results of the different evaluation methods, which were carried out on a modified version of a car data set for buying used cars [20] which was applied to illustrate our approaches given that there are no missing values in this data set. We used 300 records of used cars to generate off-line recommendation experiments. We selected car features using principal components analysis and assigned a sentiment value to each value in a feature. These values were approximated by a domain expert in the car business to simulate real data. We used equal weights to measure the local similarity, i.e., w_i in Eq. (1) was set to 1 for all features.

5.1 Case Structure

The car case base was structured in two models: CBR case model and product recommendation (PR) case model. The former is a traditional CBR case model with a problem component (car's features) and a solution component (cars' manufacture), as shown in Table 3 whereas the latter is a PR case model with one problem component (cars' features), as shown in Table 4 [15]. Both case models used a weighted sum metric to measure the global similarity at the case level between a user's query case and a candidate item case as shown earlier in Eq. (1), where weights (w_i) are equal and set to 1 for all features. The local similarity at the feature level is shown in Tables 3 and 4 for each case structure.

5.2 Query Structures and Retrievals

An example of a user query (problem component) is presented for each case structure, as seen in Fig. 4. We used these queries to calculate the ranking scores in Eq. (6) and

Fig. 4 Query structure of car data (left picture: using the PR case model; right picture: using the CBR case model)**Table 5** Top 5 retrieved cases of a selected query using different sentiment approaches of case-based recommenders

CBR Case Model					
Without Sentiment		Feature-Level Sentiment		Product-Level Sentiment	
ID	Score	ID	Score	ID	Score
6	1	6	1	6	0.74
23	0.75	23	0.89	229	0.68
27	0.75	41	0.88	278	0.68
30	0.75	96	0.87	149	0.68
41	0.75	165	0.87	48	0.68
Product Recommender Case Model					
Without Sentiment		Feature-Level Sentiment		Product-Level Sentiment	
ID	Score	ID	Score	ID	Score
6	0.88	6	1	6	0.96
23	0.66	23	0.89	76	0.75
27	0.66	41	0.88	151	0.67
30	0.66	96	0.87	229	0.66
41	0.66	165	0.87	278	0.66

Eq. (7) at the feature level and product level respectively, as illustrated in Table 5 where the w was tuned to 0.10. In our experiments, we found that better results were obtained when w was tuned to 0.1 or less.

5.3 Evaluation Analysis and Discussion

We used different methods to evaluate the performance of our sentiment case-based recommender. All of them use leave-one-out cross-validation to simulate the human-computer dialog process. This validation method uses all the cases as queries. It runs in rounds for each case in the case base. In each run, one case is used as a query, i.e., a chosen case from the case base is taken to be the user's target. The traditional case-based recommender evaluates the degree of a product's similarity with the products in the target case and not only its presence or absence [3]. This will improve the performance if the data set contains missing values. We applied our experiments using different case structures, different numbers of similar retrieved cases (k), and multi-level sentiment approaches. The first evaluation measured the accuracy of our case-based recommender as the average similarity of all cases with good solution. The similarity was measured using different equations according to the chosen recommendation approach; Eq. (1) was

Fig. 5 Evaluation results of the majority voting and weighted average voting methods of the CBR case model in our case base recommender

Table 6 Evaluation results of the accuracy-based CBR

Approach	Without Sentiment	Feature-Level Sentiment	Product-Level Sentiment
PR case model	76.33	88.49	77.907
CBR case model	76.33	84.60	74.67 (w = 0.1) 73.04 (w = 0.2)

used for the traditional approach without sentiment scores whereas the feature-level sentiment approach used Eq. (6). The product-level sentiment approach used Eq. (7). Similar approaches have been successfully used by the CBR community [21]. The results of the accuracy-based CBR are presented in Table 6. As can be seen in the table, the accuracy of both case models was highly improved with the use of feature-level sentiment approach. The second evaluation used voting methods, which provide the capability to classify a query by predicting its solution from the supplied cases. This method can be applied to the CBR case model since it requires a combination of problem-solution components. The Majority Voting (MV) method selects the predicted class that has the highest number of votes, whereas the similarity Weighted Voting (WV) method uses each vote based on the similarity of the case to the query. The class with the highest overall value is the predicted class. Figure 5 shows the evaluation results for the voting methods using different sentiment levels. In both voting methods, the accuracy rates in the feature-level sentiment approach were slightly higher than in the other approaches, whereas the accuracy rates in the product-level sentiment approach were almost the same as in the usual approach without sentiment scores.

The third evaluation used recall and precision metrics which are widely known in the information retrieval community, with more emphasis on precision. Precision is a proportion of the retrieved cases that are relevant. Meanwhile, recall is the proportion of cases relevant to the query that were successfully retrieved. In our recommender, we adapt this metric to a specific similarity threshold (70%) to measure the relevance of the retrieved cases. Precision was calculated as the number of selected and relevant cases divided by the number of selected cases (k), whereas recall was calculated as the number of selected and relevant cases divided by the number of relevant cases that had high similarity (above %70). These two metrics are, however, often

Fig. 6 IR evaluation results of our case-based recommender for different case models and sentiment-levels: WS, without sentiment; FS, feature-level sentiment; PS, product-level sentiment.

contradictory in nature. For example, decreasing the number of selected cases (k) leads to a decrease in the recall and to an increase in the precision. The F1-measure is another IR metric that provides equal weights to both recall and precision, as illustrated in Eq. (8) [22].

$$F1 = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (8)$$

To measure the effectiveness of our recommender retrieval method using the sentiment scores with the cars' descriptions, we computed the average precision, average recall and F1 measures based on them for all cases in the case base. A higher precision implied that the probability of the retrieved cases being relevant was high. Unfortunately, our data set has few relevant cars, which explains the low recall and F1-measure rates. Therefore, we focused more on the precision metric as most researchers did [3], [7], [23]. Regarding the number of similar cases that would be used, we conducted recommendations using 1, 3 and 5 similar cases. The experimental results indicated that the best option was to use only the most similar cases when k was set equal to 1. Similarly, we also found that w is best to set to 0.1 or less. Figure 6 displays the IR evaluation results for different case

models and sentiment-levels, where w was set to 0.1 in the product-level sentiment approach. As can be observed in the figure, the accuracy and precision of both case models were highly improved with the use of the feature-level sentiment approach. In the PR case model, the product-level sentiment approach outperformed the no-sentiment approach in accuracy and precision metrics while maintaining the same performance in the other CBR case model.

6. Conclusion

Recommender systems have become an important part of various commercial fields, helping customers and suppliers in their decision-making processes regarding buying and selling products. They have been developed using CBR, which enables them to recommend less popular items and generate recommendations to new users, reducing the cold-start and the overspecialization problems of collaborative filtering and content-based approaches. This approach has proven to be very successful in many e-commerce sites, owing to the clear organization of users' needs and preferences. However, case-based recommendation approaches are limited in their usage of unified case descriptions centered on technical features. This type of case representation is not appropriate in any experiential context. On contrary, cases generated from customer reviews are experiential. They can be produced from the product features that customers consider in their reviews. Moreover, these product reviews have a significant influence on the purchasing behavior of users [6].

To cope with the increased importance of user reviews, we propose an enhanced approach in case-based recommender that can utilize the opinion data from users' reviews, which are generated from the users' experience with the product, as a basis of product information. Our approach adapts the use of sentiment scores at the feature level and product level along with feature similarity throughout the recommendation. Thus combining user experience and similarity measures improves the recommender performance as illustrated in our experiments. In addition, this gives users more flexibility to choose whether they prefer products more similar to their query or better qualified products. We presented our experimental results using different evaluation methods for different case structures, different numbers of similar cases retrieved and multilevel sentiment methods. According to the experimentation done, the proposed approach performs better than the usual recommendation techniques that do not use sentiment scores. The recommender performance was highly improved with the use of the feature-level sentiment approach, which recommends products that are similar to the query but favorable to customers. This work can be further extended to use more data sets from commercial suppliers such as Amazon and include opinion mining to complete the overall approach of case-based RSs based on customers' reviews.

Acknowledgments

This work was supported by the Deanship of the Scientific Research and Research Center of the College of Computer and Information Sciences, King Saud University.

References

- [1] B. Smyth, "Case-Based Recommendation," *The Adaptive Web*, pp.342–376, Springer, 2007.
- [2] D. Bridge, M.H. Göker, L. McGinty, and B. Smyth, "Case-based recommender systems," *Knowl. Eng. Rev.*, vol.20, no.3, p.315, Sept. 2005.
- [3] A. Gatzoura and M. Sanchez-Marre, "A Case-Based Recommendation Approach for Market Basket Data," *IEEE Intell. Syst.*, vol.30, no.1, pp.20–27, Jan. 2015.
- [4] R. Dong, M. Schaal, M.P. O'Mahony, K. McCarthy, and B. Smyth, "Opinionated product recommendation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, LNAI, vol.7969, pp.44–58, 2013.
- [5] R. Dong, M.P. O'Mahony, and B. Smyth, "Further Experiments in Opinionated Product Recommendation," vol.8765, pp.108–122, 2014.
- [6] "The Impact of Online Reviews on Customers' Buying Decisions," <https://goo.gl/EEbK9c>, accessed March 29, 2016.
- [7] L. Chen, G. Chen, F. Wang, L. Chen, G. Chen, and F. Wang, "Recommender Systems Based on User Reviews: The State of the Art," *User. Model. User-Adap.*, vol.25, no.2, pp.99–154, 2015.
- [8] K.-D. Althoff and R.O. Weber, "Knowledge management in case-based reasoning," *Knowl. Eng. Rev.*, vol.20, no.3, p.305, Sept. 2005.
- [9] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, eds., *Recommender Systems Handbook*, vol.532, Springer, pp.73–105, 2011.
- [10] C.-S. Wang and H.-L. Yang, "A recommender mechanism based on case-based reasoning," *Expert Syst. Appl.*, vol.39, no.4, pp.4335–4343, 2012.
- [11] F. Lorenzi and F. Ricci, "Case-based recommender systems: A unifying view," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, LNAI, vol.3169, pp.89–113, 2005.
- [12] D. McSherry, "Diversity-Conscious Retrieval," *Advances in Case-Based Reasoning, Lecture Notes in Computer Science*, vol.2416, pp.219–233, 2002.
- [13] A. Stahl, "Combining case-based and similarity-based product recommendation," *Adv. Case-Based Reason. Proc.*, vol.4106, pp.355–369, 2006.
- [14] Q.N. Nguyen and F. Ricci, "Conversational Case-Based Recommendations Exploiting a Structured Case Model," *Advances in Case-Based Reasoning*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp.400–414, 2008.
- [15] A. Stahl, "Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning," *Dfki.Uni-Kl.De*, 2003.
- [16] D. Mcsherry, "Similarity and Compromise," *Proc. 5th International Conference on Case-Based Reasoning, ICCBR 2003*, pp.291–305, 2003.
- [17] S. Vaishnavi and S. Karthik, "Design of Recommender System Based on Customer Reviews," *International Journal of Research in Engineering and Technology*, vol.2, no.11, pp.428–431, 2013.
- [18] R. Keshav, A.S.S.R. Goutham, and J. Naren, "Content based Recommender System on Customer Reviews using Sentiment Classification Algorithms," *Int. J. Comput. Sci. Inf. Technol.*, vol.5, no.3, pp.4782–4787, 2014.
- [19] "COLIBRI | GAIA - Group of Artificial Intelligence Applications,"

- <http://gaia.fdi.ucm.es/research/colibri>, accessed May 3, 2016.
- [20] "Car data set," http://www.mycbr-project.net/downloads/used_cars_flat.csv, accessed Feb. 2, 2016.
- [21] W. Cheetham and J. Price, "Measures of Solution Accuracy in Case-Based Reasoning Systems," P. Funk and P.A. González Calero, eds. Springer Berlin Heidelberg, Berlin, Heidelberg, vol.3155, pp.106–118, 2004.
- [22] J.T.R. Badrul, M. Sarwar, George Karypis, Joseph A. Konstan, "Application of Dimensionality Reduction in Recommender System – A Case Study," Proc. ACM WEBKDD Workshop, June 2000.
- [23] S. Tyagi and K.K. Bharadwaj, "A Hybrid Recommender System Using Rule-Based and Case-Based Reasoning," Int. J. Inf. Electron. Eng., vol.2, no.4, pp.586–590, 2012.

Mashaël Aldayel received her B.S. and M.S. degrees in Information Technology and Information System from King Saud University (KSU) in 2010 and 2013, respectively. She works as a teaching assistant in Riyadh, Saudi Arabia. Currently, she is pursuing her PhD degree at KSU.

Mourad Ykhlef is an associate professor in the Department of Information Systems, KSU, Saudi Arabia. He got his Ph.D. in Computer Science from university Bordeaux 1 (France). His M.Sc. degree was in Artificial Intelligence from Paris 13 (France). He got B. Eng in Computer Science from Constantine University. His main research interests include Data Mining and Bio-inspired computing.