User Modeling and User-Adapted Interaction (2019) 29:381–441
https://doi.org/10.1007/s11257-018-9214-9

A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews

María Hernández-Rubio[1] · **Iván Cantador**[2] · **Alejandro Bellogín**[2]

Received: 15 December 2017 / Accepted in revised form: 15 November 2018 /
Published online: 24 November 2018

Abstract

In popular applications such as e-commerce sites and social media, users provide online reviews giving personal opinions about a wide array of items, such as products, services and people. These reviews are usually in the form of free text, and represent a rich source of information about the users' preferences. Among the information elements that can be extracted from reviews, opinions about particular item *aspects* (i.e., characteristics, attributes or components) have been shown to be effective for user modeling and personalized recommendation. In this paper, we investigate the aspect-based top-*N* recommendation problem by separately addressing three tasks, namely identifying references to item aspects in user reviews, classifying the sentiment orientation of the opinions about such aspects in the reviews, and exploiting the extracted aspect opinion information to provide enhanced recommendations. Differently to previous work, we integrate and empirically evaluate several state-of-the-art and novel methods for each of the above tasks. We conduct extensive experiments on standard datasets and several domains, analyzing distinct recommendation quality metrics and characteristics of the datasets, domains and extracted aspects. As a result of our investigation, we not only derive conclusions about which combination of methods is most appropriate according to the above issues, but also provide a number of valuable resources for opinion mining and recommendation purposes, such as domain aspect vocabularies and domain-dependent, aspect-level lexicons.

Keywords Recommender systems · Aspect-based recommendation · Sentiment analysis · Opinion mining · Aspect extraction · User reviews

✉ Iván Cantador
ivan.cantador@uam.es

María Hernández-Rubio
maria.hrubio@gmail.com

Alejandro Bellogín
alejandro.bellogin@uam.es

[1] BBVA Data and Analytics, Madrid, Spain

[2] Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, Madrid, Spain

1 Introduction

1.1 Motivation

In the predominant view, addressing situations of information overload and helping in decision making tasks, recommender systems aim to identify and suggest information items (e.g., products, services and people) of "relevance" for a target user (Jannach and Adomavicius 2016). Broadly, the relevance of an item can be estimated according to items the user liked in the past—*content-based (CB) recommendations*—or considering items preferred by like-minded people—*collaborative filtering (CF) recommendations*—(Adomavicius and Tuzhilin 2005).

In addition to contextual data (Adomavicius and Tuzhilin 2015), recommender systems mainly generate item relevance predictions based on both user/item attributes and user preferences, i.e., interests, tastes or needs. Such preferences are explicitly stated by the users or are inferred from past user-item interactions, commonly numeric evaluations (a.k.a. ratings) (Herlocker et al. 1999) and consumption records (Hu et al. 2008), respectively. There are, however, many popular applications—such as e-commerce sites and social media—where users not only evaluate items through ratings, but also provide personal reviews supporting their preferences.

Reviews are usually in the form of textual comments that express the reasons for which the users like or dislike the evaluated items. They thus represent a rich source of information about the users' preferences, and can be exploited to build fine-grained user profiles and enhance personalized recommendations. In this sense, Chen et al. (2015) identify various elements of valuable information that can be extracted from user reviews and can be utilized by recommender systems, namely frequently used terms, discussed topics, overall opinions about reviewed items, specific opinions about item features, comparative opinions, reviewers' emotions, and reviews helpfulness.

Frequently used terms can be used to characterize the reviewers with term-based profiles, which e.g. could be leveraged to a CB recommender (García Esparza et al. 2011). Their relevance may be determined with a weighting measure such as TF-IDF. *Discussed topics* can be utilized to enhance ratings in CF, as done in Seroussi et al. (2011). They may be obtained by grouping frequently occurring nouns or via a topic modeling technique such as Latent Dirichlet Allocation, LDA (Blei et al. 2003). The users' *overall opinions* (i.e., positive or negative sentiment orientations) about the reviewed items could be converted into virtual ratings, which may be valuable for improving CF approaches (Poirier et al. 2010; Pero and Horváth 2013; Zhang et al. 2013). They could be inferred by aggregating the sentiments of all opinion words in the reviews or via machine learning techniques. The users' *opinions about item features* can be used to enhance item profiles and increase recommendation ranking quality (Aciar et al. 2007; Yates et al. 2008; Dong et al. 2013), as latent preference factors in model-based CF (Jakob et al. 2009; Wang et al. 2012; Chen et al. 2016), and to weight user preferences in augmented recommendations (Liu et al. 2013; Chen and Wang 2013, 2014). In general, they correspond to nouns and noun phrases frequently occurring together with nearby adjectives. *Comparative opinions*, which indicate whether an item is superior or inferior to another with respect to certain

feature, can be extracted via linguistic rules. They may be used to build a graph of comparative relationships between items. Such a graph could be exploited to improve the quality of item rankings (Li et al. 2011; Jamroonsilp and Prompoon 2013; Kumar et al. 2015). The *reviewers' emotions* and *mood* (e.g., happiness, sadness) when writing the reviews can be used to determine the probability that the users will like the items, as presented by Moshfeghi et al. (2011) and Zhang et al. (2013). Finally, the *reviews helpfulness*, established in terms of the number of votes given by users to reviews, can be used to identify quality ratings that allow making better item relevance predictions (Raghavan et al. 2012).

Among the previous elements, opinions and sentiments expressed by users in personal reviews about specific features or *aspects* (i.e., characteristics, attributes or components) of the reviewed items have shown to be effective for user modeling (Wang et al. 2010; Ganu et al. 2013; Wu et al. 2015). For instance, let us consider a user who rated a particular mobile phone with an overall rating of 4 stars in a 1–5 star scale. With no more information, it is not possible to know why she gave that score instead of the highest 5-star rating. In contrast, analyzing a review she would have written about the phone, we may find out that the user thought the phone *camera* was the best she had ever used and its *battery life* was relatively long. Moreover, we could also discover that the user perceived the phone a bit heavy and quite expensive, referring to the phone *weight* and *price* respectively. These opinions about aspects of the phone are the reasons for the 4-star rating, and provide a fine-grained representation of the user's preferences.

Aspect-based recommender systems, a.k.a. recommender systems based on feature preferences (Chen et al. 2015), aim to exploit such particularities, and provide personalized recommendations taking into account the users' opinions about aspects of the rated items. Following the previous example, let us now consider a reviewer who is usually concerned about the audio characteristics of electronic devices; a fact that has been somehow inferred and incorporated into the user's profile. For this user, an aspect-based recommender system may find as more relevant and may suggest those phones that have been evaluated as having a good *voice call quality* in others' reviews. In this way, even when items are evaluated with the same rating value, these systems are able to capture particular strengths and weaknesses of the items and, based on this information, better estimate the relevance of such items for the target user, as recently shown by Bauman et al. (2017) and Musto et al. (2017).

Despite these benefits, aspect-based recommender systems have received limited attention in the research literature, even when the extraction of opinions about item aspects from user reviews is a major research topic in the area of Sentiment Analysis and Opinion Mining (Liu and Zhang 2012; Rana and Cheah 2016). Chen et al. (2015) presented an exhaustive survey on review-based recommender systems in general, and aspect-based recommender systems in particular. As shown in that survey, the majority of published papers propose recommendation approaches that follow a specific aspect extraction method, and do not evaluate existing alternatives. In most cases, the proposed recommendation approaches are empirically compared with standard user/item-based CF and matrix factorization (MF), but not with other aspect-based recommenders. Moreover, in general, reported evaluations were conducted on single domains and datasets, and using rating prediction metrics, which are progressively in

disuse and are replaced by ranking-based and non accuracy metrics. In this context, to the best of our knowledge, there is no study that clarifies which aspect extraction methods and subsequent recommendation approaches could represent the best solution for a given domain, in terms of heterogeneous recommendation quality measures.

Aiming to shed light on this situation, in this paper we separately address three tasks, namely *aspect extraction*, i.e., identifying references to item aspects in user reviews, *aspect opinion polarity identification*, i.e., classifying the sentiment orientation/polarity (e.g., as positive, neutral or negative) of the opinions about the aspects identified in the reviews, and *aspect-based recommendation*, i.e., exploiting the extracted aspect opinion information to provide enhanced personalized top-*N* recommendations. In both the aspect extraction and aspect-based recommendation tasks, we empirically compare several state-of-the-art and novel approaches on various domains and standard datasets, analyzing distinct metrics. Moreover, in the aspect opinion extraction task, we use popular natural language processing and opinion mining resources to enhance techniques on sentiment orientation identification. In particular, we consider domain-dependent aspect-level polarities of adjectives (e.g., *low price* vs. *low battery life*), adverbs modifying or intensifying such polarities (e.g., *quite/too/absolutely cheap battery*), and negation of adjectives (e.g., *non cheap battery*) and sentences (e.g., *I do not think the battery is cheap*).

As a result of our investigation, we do not only report and analyze extensive results on which combination of aspect extraction and recommendation methods may be the most appropriate for a certain domain, but also provide a number of resources valuable for researchers and practitioners, specifically, domain aspect vocabularies, domain-dependent, aspect-level lexicons (specifically, lists of positive and negative adjectives), and aspect opinion annotations of the datasets.

1.2 Research questions

In this paper, we aim to give well argued answers to the following three research questions:

– **RQ1** Is there an aspect extraction method that generates data consistently effective for both content-based and collaborative filtering strategies?

To address this question, we experiment with several state-of-the-art methods to aspect (opinion) extraction, evaluating the different types of existing approaches, namely exploiting aspect vocabularies, word frequency distributions (Caputo et al. 2017), syntactic relations (Qiu et al. 2011), and topic models (McAuley and Leskovec 2013). We integrate each of these techniques with a number of content-based and collaborative filtering methods for aspect-based recommendation. In this way, we aim to show whether combining aspect opinions and ratings as user preferences entails better recommendations, and to identify aspect extraction approaches that generate valuable data for all/most of the evaluated recommenders.

– **RQ2** To what extent are opinions about item aspects valuable to improve the quality of personalized recommendations?

To address this question, we empirically compare the developed aspect-based recommendation methods against state-of-the-art recommenders that do not exploit aspect opinion information, and HFT (McAuley and Leskovec 2013), a matrix factorization model that considers hidden topics as a proxy for item aspects. Differently to previous work, in this paper, we analyze not only the recommendation accuracy (by means of precision, recall and nDCG ranking-based metrics), but also the achieved trade-off between accuracy and other recommendation quality metrics, such as coverage, diversity and novelty.

– **RQ3** How do the coverage and type of extracted aspects affect the performance of aspect-based recommendation methods?

To address this question, we investigate scenarios with different levels of aspect opinion annotation coverage, measured in terms of the percentage of the rated/reviewed items that contain aspect opinions (identified by the developed extraction methods). We thus aim to show whether the achieved recommendation performance on the original datasets is comparable to that achieved in situations where there are aspect opinion annotations for all items. Moreover, we compare the types of aspects extracted with each method with respect to their effectiveness for improving recommendation performance and to their adequacy for explaining generated recommendations.

For the three research questions, we conduct our evaluations on popular Yelp[1] and Amazon[2] (McAuley and Yang 2016) datasets, considering user reviews about items in eight domains: hotels, beauty and spas and restaurants, and movies, digital music, CDs and vinyls, mobile phones and video games, respectively.

1.3 Contributions

In contrast to previous work, in this paper we extensively evaluate combinations of distinct methods to extract item aspect opinions from user reviews, and methods that exploit such opinions to provide personalized item recommendations. As a result of our investigation, in addition to the answers provided to the stated research questions, we claim the next contributions:

– To the best of our knowledge, we present the first empirical comparison of aspect opinion extraction methods covering the existing types of approaches, namely vocabulary-, word frequency-, syntactic relation-, and topic model-based approaches.
– We present a novel technique to estimate the sentiment orientation of opinions, which adapts the polarity of adjectives by considering adverbs that modify the intensity of the opinions, and by identifying negations of adjectives and/or sentences.
– We evaluate content-based and collaborative filtering state-of-the-art and novel aspect-based recommendation methods on several domains and well-known datasets, using heterogeneous metrics of recommendation quality, such as ranking accuracy, catalog coverage, and item novelty and diversity.

[1] Yelp Challenge dataset, https://www.yelp.com/dataset/challenge.

[2] Amazon reviews dataset, http://jmcauley.ucsd.edu/data/amazon.

Besides these contributions, we provide new categorizations and up-to-date surveys on aspect opinion extraction and aspect-based recommender systems. Moreover, we make publicly available[3] the following resources:

- Aspect-level lexicons with the polarity of adjectives associated to item aspects in reviews for the addressed domains.
- Vocabularies composed of nouns appearing in user reviews that refer to aspects for the above domains.
- Lists of weighted adverbs that strengthen, soften or invert the polarity of adjectives.
- Aspect opinion annotations of the used datasets, which are popular in the Sentiment Analysis and Opinion Mining research area.

1.4 Structure of the paper

The remainder of the paper is structured as follows. In Sect. 2, we revise related work on aspect opinion extraction and aspect-based recommendation, following formal categorizations of existing approaches for both tasks. Selected as state-of-the-art examples from each of the identified categories or proposed as novel methods, in Sects. 3 and 4, we describe the developed and integrated aspect extraction techniques and aspect-based recommenders. Next, in Sects. 5 and 6, we present the experiments conducted to address the stated research questions, describing the experimental setting and analyzing the achieved empirical results, respectively. Finally, in Sect. 7 we end with some conclusions and future research lines.

2 Related work

In this section, we survey the research literature on the two main tasks involved in the aspect-based recommendation problem, namely extracting opinions about item aspects from user reviews (Sect. 2.1), and exploiting the extracted opinion information for personalized item ranking (Sect. 2.2).

2.1 Aspect opinion extraction approaches

In the subsequent subsections, we discuss state-of-the-art aspect (opinion) extraction methods, following an own categorization based on those presented by Liu (2012) and Rana and Cheah (2016). We focus on unsupervised methods, where no manually labeled aspect annotations are needed, and specifically we distinguish between the following approaches: *vocabulary-based* methods that make use of lists of aspect words (Sect. 2.1.1), *word frequency-based* methods in which words that have a high appearance frequency are selected as aspects (Sect. 2.1.2), *syntactic relation-based* methods where syntactic relations between words of a sentence are the basis for identifying aspect opinions (Sect. 2.1.3), and *topic model-based* methods where topic models are used to extract the main aspects from user reviews (Sect. 2.1.4). Next, in Sect. 2.1.5,

[3] Aspect opinion resources, http://ir.ii.uam.es/aspects.

we compare the surveyed methods and analyze their strengths and weaknesses. Differently to Liu (2012), we exclude aspect extraction methods based on supervised learning (Jakob and Gurevych 2010) since they rely on large amounts of labeled data, an uncommon scenario in real applications. Moreover, in contrast to Rana and Cheah (2016), we do consider topic modeling techniques as they have been proven to be very effective in representing item aspects from reviews (Titov and McDonald 2008b; Zhao et al. 2010a; McAuley et al. 2012; Diao et al. 2014).

In addition to the way in which references to aspects are identified in user reviews, it is important to describe how the sentiment orientation or polarity of the opinions about aspects is established. In this context, at some point, existing solutions make use of lexicons. In the simplest form, a sentiment/opinion lexicon (or simply lexicon) is composed of lists of adjectives that are used to reflect *positive* or *negative* subjectivity characteristics or qualities of any type of entity. There are lexicons that contain other types of words (e.g., nouns, adverbs and verbs), lexicons that provide numeric polarity scores (e.g., in a $[-5, 5]$ range), and lexicons that include misspellings, morphological variants, slang expressions, and social media mark-up. In general, available lexicons are limited to words that express generic, domain-independent subjectivity. We will cite which lexicons are used in the papers surveyed.

2.1.1 Vocabulary-based extraction

The most direct approach to identify aspect opinions in reviews is by means of a vocabulary with the terms that refer to aspects. Aciar et al. (2007) presented a semi-automatic method that identifies references to aspects in user reviews through an ontological structure. When processing user reviews, each sentence that contains words mapped to an aspect ontology is annotated with the corresponding ontology concepts. Afterwards, a text mining technique is used to select and classify a review sentence as *good* or *bad* if it contains information about features that the user has evaluated as item strengths and weaknesses, respectively. The method thus needs an initial, domain-dependent ontology manually built in advance, whereas its annotation algorithm is fully automatic.

2.1.2 Word frequency-based extraction

One of the simplest, yet effective, approaches to extract references to aspects from textual reviews consists of identifying words frequently used in a specific domain. In this context, Hu and Liu (2004a) presented a method aimed to summarize textual reviews, highlighting the fragments most valuable for readers according to their information needs. Specifically, the authors used association rule mining and the *Apriori* algorithm (Agrawal et al. 1994) over nouns and noun phrases to find frequent *itemsets*, and performed a pruning stage to keep only the most informative ones, which are assumed to refer to evaluated item aspects. In their methods, the sentiment orientation of each aspect opinion is assigned based on the nearest adjectives to the selected nouns. In particular, an aspect opinion is annotated with the polarity (or inverse polarity) that the corresponding adjective—or any of its synonyms (or antonyms) obtained from

WordNet (Miller 1995)—have in the well-known lexicon presented in Hu and Liu (2004b).

This method was improved in Popescu and Etzioni (2005) and Bafna and Toshniwal (2013) by removing those frequent nouns that are not likely to represent aspects. Specifically, Popescu and Etzioni (2005) considered that an aspect is *part* or *feature* of a product, and can be identified by means of high Point-Wise Mutual Information (PMI) values,

$$\text{PMI}(f, d) = \frac{\text{hits}(f, d)}{\text{hits}(f) \cdot \text{hits}(d)},$$

between potential aspect words f and *meronymy discriminators* d associated with the type of the product, e.g., "of phone", "phone has" and "phone comes with" for the *phone* type. For this computation, the authors utilized the *hits* statistics provided by the *KnowItAll Assessor* system (Etzioni et al. 2005), which obtains relationships such as *isPartOf(screen, phone)* by querying the Web. Bafna and Toshniwal (2013), in contrast, investigated a probabilistic approach to select all those nouns that are likely to represent aspects.

Scaffidi et al. (2007) built the *Red Opal* system, which makes use of a Language Model to identify references to aspects in reviews, and detect those that the target user is more interested in. The authors assumed that item aspects are mentioned more often in a review than in a multi-domain corpus. For instance, in a collection of reviews about restaurants, words as 'ambiance', 'service', 'food', 'dessert' or 'price' tend to appear much more often than in document repositories of other domains. Their method computes the probability that a word t is observed n_t times in a review of length N, and compare it to the ratio of appearance in standard English, p_t. If the ratio is high, then the word t is considered to be an aspect word. The opinion sentiment orientation is assigned based on the assumption that the global rating of a review correlates with the polarity of each word. *Red Opal* thus only considers the review ratings to estimate the user's interest on the items, and avoids analyzing opinion words.

Recently, Caputo et al. (2017) have presented the *SABRE* search engine, which, similarly to the *Red Opal* system, compares the word frequency distributions in a target, single-domain document collection with distributions in a general, multi-domain corpus. SABRE produces as output a set of tuples describing an input review. Such tuples contain extracted aspects together with their relevance and sentiment, along with sub-aspects related to the aspects, if exist. The key point of this method is how word relevance is measured. The authors use the point-wise Kullback–Leibler divergence (KL divergence, referred to as δ) with respect to a general corpus. Formally, given two corpora c_a and c_b, and a word t, the KL-divergence is calculated as:

$$\delta_t(c_a||c_b) = p(t, c_a) \log \frac{p(t, c_a)}{p(t, c_b)}$$

The proposed method computes the KL divergence for each of the extracted nouns on the domain and general corpus, and considers those nouns with a KL score higher than certain threshold ε to be item aspects.

2.1.3 Syntactic relation-based extraction

Another type of approach to aspect opinion extraction focuses on analyzing the syntactic sentence structure and word relations. Qiu et al. (2011) presented the Double Propagation (DP) algorithm, which exploits syntactic relations between the words in a review to identify those that correspond to aspects. More specifically, the algorithm makes use of the relations between nouns or noun phrases, and adjectives. It utilizes dependency grammar to describe such syntactic relations (Schuster and Manning 2016), and follows a set of extraction rules. Using a lexicon, the basic idea of DP is to extract opinion words or aspects by iteratively using known and previously extracted opinion words and aspects. To illustrate the algorithm, let us consider the sentences "Canon G3 takes great pictures", "The picture is amazing", "You may have to get more storage to store high quality pictures and recorded movies" and "The software is amazing," and the input positive opinion word *great*. DP first extracts *picture* as an item aspect based on its relation with *great*. Analyzing other relations of these aspect and opinion words, DP determines that *amazing* is also an opinion word, and *movies* is another aspect. In a second iteration, as *amazing* is recognized as an opinion word, *software* is extracted as an aspect. This propagation stops as no more aspect or opinion words are identified. The polarity associated to each aspect is assigned at the same stage than the extraction. It is based on the polarity of the known word that it is related to, considering negation and contrary words in the sentence. This method may propagate noise when extracting aspects terms that are not real aspects. This problem was addressed by Qiu et al. (2011), by means of a final pruning stage. The DP algorithm has become the basis of several state-of-the-art methods for extracting opinions about item aspects from textual reviews, and some works have presented improvements over the originally proposed set of propagation rules. For instance, Zhang et al. (2010) introduced "part-whole" and "no" pattern rules to identify aspects. The "part-whole" pattern extracts aspects mentioned in a review as part of another product, as in "the engine of the car," where *engine* is part of *car*. The "no" pattern handles phrases like "no noise". Poria et al. (2014) also proposed a variation of DP by extending the set of rules and accounting for verb words as aspects.

2.1.4 Topic model-based extraction

Most of the approaches analyzed in previous subsections extract a list of words referring to aspects in reviews. In this context, several words may refer to the same aspect. For example, users may talk about the *service* in a restaurant by using distinct words like 'service', 'staff' and 'attention', which should not be considered as different aspects. Aciar et al. (2007) manually handled this issue defining an ontology that groups related words. There are, in contrast, methods that rely on Topic Models, such as LDA (Blei et al. 2003) and pLSA (Hofmann 2001), for both extracting and clustering aspect-related words automatically in a single phase.

If LDA or pLSA are applied in a straightforward way, they might not be able to capture the appropriate item aspects. In particular, they tend to build general topics that map terms into concepts the reviews talk about. For example, in the restaurants domain, topics are usually related to types of cuisine, such as Italian, Asiatic, vegetarian and

vegan; in movies and books reviews, topics in general correspond to genres; and in electronics reviews, topics tend to represent different types of devices. Hence, several works have investigated particular topic models to find more fine-grained concepts in the reviews. Titov and McDonald (2008a) proposed Multi-Grain Topic Models (MG-LDA), a probabilistic approach that focuses on both *global* and *local* topics. Global topics are described by words related to the domain or general properties of the reviewed items, whereas local topics capture item aspects or features. This approach improves the quality of LDA by considering as aspects only those topics that can be explicitly rated. The same authors, in Titov and McDonald (2008b), enhanced the probabilistic model to associate the topics obtained with MG-LDA with particular item aspects. The followed method is based on the assumption that aspect ratings should be correlated with item ratings. Hence, the global rating of the review may be helpful to identify topics that correspond to aspects.

McAuley et al. (2012) presented a probabilist model that exploits the ratings associated with the reviews to simultaneously learn words that refer to aspects, and words that are associated with particular ratings. For instance, the word 'flavor' may be used to discuss the *taste* aspect, whereas the word 'amazing' may indicate a 5-star rating (in an 1–5 star scale). In the paper, the authors present three (unsupervised, semi-supervised and supervised) learning methods to build the model; in all cases, requiring a ground-truth set of ratings on aspects.

More recently, in the context of recommender systems, some works have related the representation of an item in the latent factor model (Koren et al. 2009) to the latent topics in reviews. In low-rank Matrix Factorization (MF), a user **u** and item **i** and can be respectively associated with k-dimensional latent factors $p_u, q_i \in \mathbb{R}^k$. Their rating is then estimated as $\hat{r}_{u,i} = p_u^T \cdot q_i$. These factors can be considered as item properties and the preference of the user for these properties, respectively. Based on this representation, Wang and Blei (2011) presented the Collaborative Topic Regression (CTR) model, where MF and LDA are run in the same stage. The latent item factor q_i is set to be the topic proportions in LDA θ_i plus an offset ε_i as $q_i = \theta_i + \varepsilon_i$. Thereafter, McAuley and Leskovec (2013) presented HTF, a slightly modified version of CTR in which latent topics in the reviews and latent factors for the item are related by a monotonic function (order is preserved):

$$\theta_{i,k} = \frac{\exp \kappa \, q_{i,k}}{\sum_{k'} \exp \kappa \, q_{i,k'}}$$

where κ controls the peakiness of the transformation.

2.1.5 Discussion

In the previous subsections, we have surveyed several works proposed in the last decade to extract aspects and associated opinions from textual reviews. We have categorized them according to the approaches they use to extract the aspects, and the required input data. Specifically, we have analyzed vocabulary-, word frequency-, syntactic relation- and topic model-based approaches.

Except those based on topic models, the majority of the surveyed methods do not consider that different words may refer to the same aspect. This represents the main limitation we identify in the heuristic approaches. Topic model-based techniques intrinsically solve such limitation. Instead of extracting specific words, they are able to capture and group the main topic the reviews are about. To identify which of the extracted topics represent aspects, standard LDA models are modified so different generation distributions can focus on specific parts of the reviews. Hence, the extraction procedures lead to K topics, each of them represented by a collection of aspect terms. The main weakness of this type of approach is that the output topics might not precisely represent aspects, but a mixture of aspects and global characteristics of the items. However, as we shall show in Sect. 2.2, such topics have been shown very effective when exploited by recommender systems.

In the surveyed works, most of the reported experiments have been conducted on small product datasets of less than a hundred reviews (Hu and Liu 2004a; Popescu and Etzioni 2005; Bafna and Toshniwal 2013; Qiu et al. 2011; Poria et al. 2014), and only a few of them have focused on larger datasets (Scaffidi et al. 2007; Liu et al. 2012). Moreover, in general, methods from different types are not empirically compared. As we shall present in Sect. 3, in this paper, we evaluate the surveyed types of aspect opinion extraction approaches for the aspect-based recommendation problem, by combining representative extraction methods with several recommendation algorithms. Moreover, we evaluate the considered combinations on large datasets, ranging from a few to more than a hundred thousands reviews.

2.2 Aspect-based recommender systems

In this section, we provide an exhaustive survey of the research literature on aspect-based recommender systems. In some of the analyzed papers, item *aspects* are referred to as *features* and *topics*. In fact, some of the discussed recommendation approaches—such as those based on Topic Models—consider aspects that may correspond to content-based attributes and context values. For simplicity, we always use the term *aspect*, regardless the terminology and aspect type used in the cited papers. Moreover, although being related work of interest, we omit papers presenting information filtering (Scaffidi et al. 2007), question answering (McAuley and Yang 2016), and information retrieval (Caputo et al. 2017) systems that exploit aspect opinion data.

We present the surveyed articles following an own categorization, which is defined upon the one proposed by Chen et al. (2015). Specifically, we distinguish between the following types of approaches: enhancing item profiles with aspect opinion information (Sect. 2.2.1), modeling latent user preferences on item aspects (Sect. 2.2.2), deriving user preference weights from aspect opinions (Sect. 2.2.3), and incorporating aspect-level user preferences into recommendation methods (Sect. 2.2.4). Next, in Sect. 2.2.5, we discuss limitations identified in the literature that have motivated our work.

2.2.1 Enhancing item profiles with aspect opinion information

A first type of approach to exploit item aspect opinion information for recommendation purposes focuses on building enhanced representations of items.

Aciar et al. (2007) presented a seminal work in this line. They proposed an ontological item representation with two components: an *item quality* component containing the user's evaluation of item aspects, and an *opinion quality* component including several variables that measure the opinion providers expertise with the item. The authors use text mining tools to first classify the sentences of each item review as *good*, *bad* and *quality*; the latter referring to the opinion quality component. Afterwards, the aspects mentioned in each of the classified sentences are extracted. Item profiles are then built applying a number of computations with the extracted data. In the paper, the authors propose a simple content-based recommendation model that ranks items according to both the item profiles and the user's current interest on the aspects, explicitly stated or estimated from the aspect frequencies in the user's reviews.

Yates et al. (2008) proposed an item profile that combines aspect opinions extracted from reviews and item technical specifications (e.g., a camera *lens* and *resolution*). This profile is called the item value model $V(i)$, and indicates the intrinsic value of the item i for an average user. The item prize, considered as an indicator of extrinsic value, is treated as the dependent variable in a training phase where a SVM model is built on new items to predict their intrinsic values. Assuming the existence of a personalized value model $V(u)$ for user u in the same aspect space as $V(i)$, the difference $\frac{V(u)-V(i)}{V(i)}$, change-in-value, reflects i's suitability for u. A user is then recommended with the items having the highest change-in-value scores.

Ko et al. (2011) proposed to represent an item as a vector composed of key aspects—relevant terms derived from user reviews and item descriptions—with importance and sentiment scores. The item vectors are built for each user separately from the ratings and reviews of similar users. Then, for each user, a binary (*recommendable* and *non-recommendable* items) classification model is learned from the derived vectors, and used for item recommendation.

Finally, Dong et al. (2013) presented an item profile composed of aspects, each of them with sentiment and popularity scores. They applied a shallow natural language processing technique to extract single nouns and bigram phrases as item aspects, and an opinion pattern mining method to identify the opinions given to the aspects. The authors proposed a case-based recommendation method that matches the user's profile—given as an input example item—with items whose profiles are highly similar and produce greater sentiment improvements.

2.2.2 Modeling latent user preferences on item aspects

A major approach to aspect-based recommendation consists of analyzing a user's reviews to infer latent preferences (ratings) on item aspects, and exploiting such aspect-level user preferences through collaborative filtering techniques.

The work done by Jakob et al. (2009) represents one of the first attempts to extract opinions about aspects from user reviews, and incorporate them into the Matrix Factorization (MF) model (Koren et al. 2009). The authors presented a model that captures

several types of relations between users, items and item aspects, namely user ratings, item aspects, user opinions on aspects, and rating- and aspect-based user similarities. These relations are treated as feature vectors for running the Multi-Relational Matrix Factorization (MRMF) algorithm proposed by Lippert et al. (2008). The aspects were extracted using LDA and the Subjective Lexicon (Wilson et al. 2005). Wang et al. (2010) proposed LRR, a probabilistic regression model to infer latent ratings on aspects. The model assumes that a rating on an item is generated through a weighted combination of latent ratings over all the item aspects; where the weights represent the relative emphasis the user has placed on the aspects, and an aspect latent rating depends on the review fragment that discusses such aspect. Using their model, the authors proposed a CF method that personalizes a ranking of items by using only the reviews written by the k reviewers whose aspect-level rating behavior is most similar to the target user's. A two-component approach is also presented by Wang et al. (2012) and Nie et al. (2014). In this case, the extraction of aspect opinions is performed through the Double Propagation (Qiu et al. 2011) and LDA algorithms, whereas recommendations are generated via a tensor factorization method that assembles the overall rating matrix $\mathbf{R}$ and K aspect rating matrices $\mathbf{R}^1, \mathbf{R}^2, \ldots, \mathbf{R}^K$ into a 3rd-order tensor $\mathcal{R}$, with which CF is performed. Ganu et al. (2013) proposed a clustering-oriented CF method based on aspect-level user preferences. The method first builds a SVM classifier to categorize review sentences into a fixed number of aspects (called topics in the paper) and sentiment categories. Based on the classification of the sentences of a user's reviews, the method builds the user's profile, composed of weighted (aspect, sentiment) tuples. Using the generated user profiles, a soft clustering algorithm is applied to group users with similar aspect-level preferences. The obtained user clusters are finally incorporated into the CF heuristic.

Instead of addressing the aspect-based user preference extraction and recommendation tasks separately, McAuley and Leskovec (2013) presented HFT, a matrix factorization model that incorporates hidden topics as a proxy for item aspects. The model aligns latent factors in rating data with latent factors in review texts. In this context, an identified topic may not correspond to a particular aspect or may be associated with several aspects, and thus a user may express different opinions for various aspects in the same topic. Nonetheless, the authors show that HTF predicts ratings more accurately than other models that consider either of such data sources in isolation, especially for cold-start items, whose factors cannot be fit from only a few ratings, but from a few reviews. Wu et al. (2014) presented JMARS, a probabilistic approach based on CF and topic modeling. Similarly to Wang et al. (2010), JMARS model assumes that review ratings arise from the process of combining ratings associated to aspects of the evaluated items. In contrast, JMARS jointly models user and item aspect rating distributions. In the same line of the work, Wu et al. (2015) present FLAME, an extension of Probabilistic Matrix Factorization (PMF) (Salakhutdinov and Mnih 2007) to model the user-specific aspect ratings. Finally, Chen et al. (2016) presented LRPPM, a tensor-matrix factorization algorithm that models interactions among users, items and features simultaneously, to learn user preferences from ratings along with textual reviews. Differently to previous work, the proposed method introduces a ranking-based (i.e., learning to rank), instead of a rating-based, optimization objective, for better understanding user preferences at aspect level.

2.2.3 Deriving user preference weights from aspect opinions

Another type of approach to aspect-based recommendation uses aspect opinion information to establish the weights of preferences in user profiles, rather than using it to infer such preferences. In these approaches, a user u_m's profile is represented as a vector $\mathbf{u}_m = \{w_{m,1}, w_{m,2}, \ldots, w_{m,K}\}$, where $w_{m,k}$ denotes the relative relevance (weight) of aspect a_k for u_m, and K is the total number of aspects.

In particular, Liu et al. (2013) proposed to determine the weight $w_{m,k}$ by means of two factors, namely how much the user concerns about the aspect, and how much quality the user requires for such aspect; formally, $w_{m,k} = concern(u_m, a_k) \times requirement(u_m, a_k)$. The value of $concern(u_m, a_k)$ increases when u_m comments on a_k very frequently in his/her reviews, and other users comment on it less often. The value of $requirement(u_m, a_k)$, on the other hand, increases when u_m frequently rates a_k lower than other users across different items. In the paper, the authors extract aspect opinions through a technique that accommodates to characteristics of the Chinese language. They also propose a recommendation method that estimates the relevance score $relevance(\mathbf{u}_m, \mathbf{i}_n) = \sum_{k=1}^{K} w_{m,k} \times v_{n,K} / \sum_{k=1}^{K} w_{m,k}$, where $v_{n,k}$ is the average of reviewers' opinions about aspect a_k of item i_n. The method recommends to u_n the top-N items with the highest relevance scores.

Differently to Liu et al. (2013), Chen and Wang (2013) focused on the cold-start situation where a user has not made enough reviews with which determining his/her aspect preference weights. The authors proposed a method that first derives cluster-level preferences, which are common to groups of users. Then, these cluster-level preferences are used to refine the users' personal preferences. The refined preferences can in turn be used to adjust the cluster-level preferences, continuing the process until both types of preferences do not change significantly. This method is executed on an initial set of (aspect, opinion) tuples extracted from the user reviews. In the aspect extraction stage, the authors utilized WordNet (Miller 1995) and SentiWordNet (Esuli et al. 2007) to group aspect synonyms and determine aspect opinion polarities, respectively. In the recommendation stage, all the users are first clustered according to their cluster-level preferences, and then heuristic user-based CF is applied within the cluster to which the target user belongs.

To derive the weights of a user's preferences, it could be valuable to consider his/her current contextual conditions (Adomavicius and Tuzhilin 2015). For instance, when searching for hotels, the aspects *atmosphere* and *location* may be of interest if the user wants to spend a weekend with his/her partner, whereas *cleanliness* and *price* may be the most important aspects if the user is planning one-week holidays with his/her family. In this example, *period of time* and *companion* would be the context variables that determine the current relevance (weight) of the above aspects for the user. Some researchers have investigated this issue. Levi et al. (2012) proposed to compute the preference of user u_m for aspect a_k within contexts $C = \{c_1, c_2, \ldots, c_L\}$ as $w_{m,k} = importance(u_m, a_k) \cdot \prod_{l=1}^{L} freq_{k,l}$, where $importance(u_m, a_k)$ is the current importance of aspect a_k explicitly stated by u_m, and $freq_{k,l}$ is the frequency with which aspect a_k occurs in reviews with context c_l. With this definition of aspect-level preference, the authors estimate the relevance of each review d for user u_n as $relevance(u_m, d) = \sum_{s \in S(d)} \sum_{a_k \in A(s)} w_{m,k} \cdot so(a_k, s)$, where $S(d)$ represents the

sentences of review d, $A(s)$ is the set of aspects commented in the sentence s, and $so(a, s)$ returns the sentiment orientation (polarity) of the opinion on aspect a given in sentence s. Then, the authors present a content-based recommendation method that suggests the items i with highest review relevance scores. Differently to Levi et al. (2012), Chen and Chen (2014) aimed to directly extract the relation between preference weights and context in user reviews, by considering the co-occurrences of aspect opinions and context values. They distinguished between context-independent and context-dependent user preferences. The former are identified by building a regression model for overall ratings and aspect opinions of reviews, and applying a statistical t-test to select the model weights passing a significance level; the latter are extracted through a contextual review analysis based on keyword matching, and a rule-based reasoning on contextual aspect opinion tuples. The authors finally incorporated the derived preference weights into the recommendation approach proposed by Levi et al. (2012).

2.2.4 Incorporating aspect-level user preferences into recommendation methods

A last type of approach is represented by recommendation methods that explicitly incorporate aspect-level user preferences into their heuristic functions or predictive models for item relevance estimation.

Using the Stanford CoreNLP toolkit (Socher et al. 2013), Wang et al. (2013) presented an approach that first analyzes reviews to derive user preferences for aspect values in the form of (aspect, sentiment orientation, aspect value) tuples, such as (*weight*, *positive*, 200 g) to denote that the user expressed a *positive* opinion about a camera *weight* whose value is 200 g. These tuples are then linked to item specifications $\mathbf{i}$ using the algorithm presented in Chen and Wang (2013), and compared among users in a CF fashion to derive unknown aspect-level user preferences. After estimating the target user u's preferences $\mathbf{u}(k)$ and the candidate item i weighted attributes $\mathbf{i}(k)$ on aspects a_k, the method estimates the relevance of i as $\frac{1}{K}\sum_{k=1}^{K}\mathbf{u}(k)\cdot\mathbf{i}(k)$. The top-$N$ items with highest relevance scores are finally recommended to u.

Recently, Bauman et al. (2017) presented SULM, a sentiment utility logistic model that simultaneously fits the opinions extracted from reviews and the ratings provided by the users. SULM assumes that a user u's overall level of satisfaction with consuming item i is measured by an utility value $V_{u,i} \in \mathbb{R}$. This overall utility is estimated as a linear combination of the individual (inferred) sentiment utility values $\hat{V}^k_{u,i}(\theta_s)$ for all the aspects in a review, $\hat{V}_{u,i} = \sum_k \hat{V}^k_{u,i}(\theta_s)\cdot(w^k + w^k_u + w^k_i)$, where w^k is a general coefficient expressing the relative importance of aspect a_k, w^k_u is a coefficient that represents u's individual importance of aspect a_k, and w^k_i is a coefficient that determines the importance of aspect a_k for item i. Denoting these coefficients by $\theta_r = (W_A, W_U, W_I)$, and the set of all parameters by $\theta = (\theta_r, \theta_s)$, the model estimates θ such that the a logistic transformation g of the overall utility $\hat{V}_{u,i}(\theta)$ would fit binary ratings $r_{u,i} \in \{0, 1\}$ as $\hat{r}_{u,i}(\theta) = g(\hat{V}_{u,i}(\theta))$. The model is built by searching for the θ values that maximize the log-likelihood function $l_r(R|\theta) = \sum_{u,i} r_{u,i}\cdot\log(\hat{r}_{u,i}(\theta)) + (1 - r_{u,i})\cdot\log(1 - \hat{r}_{u,i}(\theta))$. In the paper, the authors make use of the Double Propagation algorithm (Qiu et al. 2011) to extract item aspect opinions from the user reviews.

Finally, Musto et al. (2017) presented multi-criteria user- and item-based collaborative filtering heuristics that incorporate aspect opinion information. For the user-based case (the item-based case is analogous), the authors propose an aspect-based user distance calculated as $dist(u, v) = \frac{1}{|I(u,v)|} + \sum_{i \in I(u,v)} \sqrt{\sum_{a \in A(u,i) \cap A(v,i)} |r_a(u, i) - r_a(v, i)|^2}$, where $I(u, v)$ is the set of items rated by both users u and v, $A(u, i)$ is the set of aspects commented in user u's review about item i, and $r_a(u, i)$ is the sentiment rating inferred for aspect a in that review. The similarity between users is then calculated as the opposite of the distance d, and ratings are computed through the traditional CF heuristic, $\hat{r}(u, i) = \sum_{v \in N(u)} sim(u, v) \cdot r(v, i) = \sum_{v \in N(u)} (1 - dist(u, v)) \cdot r(v, i)$, where $N(u)$ is u's neighborhood with his/her most similar users. In the paper, the aspect extraction is performed with the SABRE engine (Caputo et al. 2017).

2.2.5 Discussion

In the previous subsections, we have surveyed more than 20 research papers on aspect-based recommender systems published in the last decade, categorizing them according to how they model and weight user preferences at aspect level, and how they incorporate such preferences into the recommendation generation process. For most cases, we have seen that the aspect extraction and aspect-based recommendation tasks are addressed separately. In general, however, in each paper, only one aspect extraction method is performed, without assessing existing alternatives. To the best of our knowledge, only Musto et al. (2017) tested the SABRE engine (Caputo et al. 2017) with two sentiment analysis strategies: a deep learning technique provided by the Stanford CoreNLP toolkit and a lexicon-based algorithm evaluated in Musto et al. (2014), finding no significant performance differences between them. As explained in Sect. 3, in this paper, we shall evaluate several aspect extraction methods, each of them belonging to one of the approaches types presented in Sect. 2.1.

Moreover, in most cases, the proposed recommendation approaches were empirically compared with standard baselines that do not exploit aspect opinions, but overall item ratings; in this context, a few exceptions exist, such as Wu et al. (2014) and Bauman et al. (2017), where JMARS and SLUM where evaluated against HTF (McAuley and Leskovec 2013). As explained in Sect. 4, in this paper, additionally to standard rating-based baselines and HFT, we shall evaluate a number of content-based and collaborative filtering methods that exploit aspect opinion data.

We finally note that in many studies, the reported experiments were conducted on small datasets, for one or a few domains, and using rating prediction metrics (MAE, MSE, RMSE), which are in relative disuse within the recommender systems community (Bellogín et al. 2011). In this paper, as presented in Sects. 5 and 6, we shall run our experiments on two review-oriented datasets from Yelp and Amazon, as in Levi et al. (2012), McAuley and Leskovec (2013), Socher et al. (2013), Wang et al. (2013), Chen et al. (2016), Musto et al. (2017) and Bauman et al. (2017), covering 8 domains: hotels, beauty and spas, restaurants, movies, digital music, CDs and vinyls, mobile phones, and video games. Instead of rating prediction metrics, as done e.g. by Levi et al. (2012), Liu et al. (2013), Chen et al. (2016) and Bauman et al. (2017), we will

compute ranking-based metrics, focusing on the top-N recommendation task. Differently to previous work, we will also analyze other metrics measuring recommendation coverage, diversity and novelty.

3 Developed aspect opinion extraction methods

In this section, we present the evaluated methods to aspect opinion extraction. We have selected a representative method for each type of approach described in Sect. 2.1, namely vocabulary-, word frequency-, syntactic relation-, and topic model-based approaches. As we shall show in our experimental study (Sects. 5 and 6), when applicable, we will integrate each of the developed aspect opinion extraction methods with several content-based and collaborative filtering techniques, described in Sect. 4.

3.1 Vocabulary-based method

Our first aspect opinion extraction method makes use of a vocabulary for item aspects on a specific domain, and analyzes syntactic relations between the words of each sentence to extract opinions about aspects. The vocabulary contains a predefined list of item aspects, and a fixed set of nouns referring to each aspect, e.g., 'staff', 'employees', 'waiters' and 'waitresses' for the *staff* aspect of the *restaurants* domain. The method searches for the vocabulary nouns cited in the text of an input review, and for each of the found nouns, it generates an aspect annotation. Next, it builds the annotation in the form of a $(u, i, a, so_{u,i,a})$ tuple, where $so_{u,i,a}$ is the sentiment orientation of the opinion given by user u to aspect a of item i—usually represented by a numeric value that is lower than, equal to, or greater than 0 when the opinion is negative, neutral or positive, respectively. In this context, for a given dictionary, the followed method differs from others in the way the sentiment orientation so is determined. From now on, we will refer to our method as **voc**. All the resources created for and generated by this method, and presented next, are publicly available.[3]

3.1.1 Aspect vocabulary building

A vocabulary used by the **voc** method is composed of lists of nouns that refer to item aspects on a particular domain. We manually selected the aspects, including those that have been considered in research papers (Sect. 2), and those that correspond to item features, attributes and characteristics reported or analyzed in specialized forums (e.g., e-commerce sites, product review web portals), such as AllMusic[4] for music and GameSpot[5] for video games, among others. The selection of some of these aspects have to be carefully done in certain domains. For instance, in the restaurant domain, we observed that there were reviews with opinions about dishes focused on particular principal ingredients, such as 'rice' and 'potatoes'. We assumed that people may find valuable reviews about dishes and restaurants that received positive opinions on those

[4] AllMusic record reviews, https://www.allmusic.com.

[5] GameSpot Video Games reviews and news https://www.gamespot.com.

ingredients. We also decided to include them since topic model-based methods identify such aspects in user reviews.

Next, for each aspect, we created an initial list of 'seed' words, corresponding to the WordNet[6] (Miller 1995) synonyms of the aspect names, e.g., 'atmosphere', 'ambiance' and 'ambience' for the 'atmosphere' domain. We then extended each list with synonyms of the obtained seeds, since the name chosen for an aspect in the vocabulary may not have all the valid synonyms in WordNet. For a particular word, we only considered the synonyms of the WordNet *synsets* (i.e., word meanings) whose definitions contained certain reference word of the target domain, e.g., 'music' and 'movies'. Thus, we limit the number of obtained synonyms, but avoid ambiguities. In the list, we also included plural forms of the seed nouns, and morphological deviations of seed compound nouns, e.g., 'checkin', 'check-in' and 'check in' in the *hotels* domain.

Finally, we automatically searched for all the obtained aspect nouns in a large collection of text reviews (about items in the target domain), scoring each noun with the number of reviews in which it occurred. Merging singular, plural and compound forms of the found nouns, we sorted them by decreasing scores. We then filtered out those nouns with a score lower than certain threshold, established for each aspect and domain by manual inspection.

Table 1 shows the 8 generated aspect lists to be exploited by the **voc** method in our experiments. The table shows the aspects considered for each domain, and the number of aspect nouns compiled in each vocabulary. On average, a vocabulary has 29.7 aspects and 296.4 nouns, i.e., 10 nouns per aspect approximately. After a careful inspection of the used reviews, we claim that no or only few additional relevant aspects or aspect words can be found in our datasets. Thus, we believe that experiments and results reported in this paper are correct in that respect.

3.1.2 Aspect opinion extraction

To extract opinions about item aspects from user reviews, the **voc** method first identifies in a review occurrences of any noun stored in the aspect vocabulary of the target domain. If an occurrence is found, the method analyzes the sentence in which the noun appears, in order to obtain a potential opinion about the corresponding aspect.

For such purpose, similarly to previous work (e.g., Wang et al. 2013; Caputo et al. 2017), our method makes use of the Stanford CoreNLP toolkit (Socher et al. 2013) to language natural processing; specifically, its Part-of-Speech (POS) tagger (Toutanova et al. 2003) and syntactic dependency parser (Chen et al. 2014). On a given sentence, the POS tagger returns the Penn Treebank[7] POS tag of each word, e.g., NN, NNS, NNP and $NNPS$ for singular/plural common/proper nouns, and JJ, JJR and JJS for positive/comparative/superlative adjectives. The syntactic dependency parser, on the other hand, returns binary grammatical dependencies in the sentence as a list of `(gov, rel, dep)` triples, representing the relations `rel` hold between governors `gov` and dependents `dep`. The parser current representation contains approximately

[6] WordNet lexical database, https://wordnet.princeton.edu.

[7] Penn Treebank, http://web.mit.edu/6.863/www/PennTreebankTags.html.

Table 1 Domain-dependent aspect vocabularies used in the experiments

Vocabulary	Aspects
Hotels (31, 302)	Staff, bedrooms, bathrooms, location, building, pool, service, food, breakfast, price, bar, restaurant, atmosphere, dinner, checks, drinks, events, amenities, facilities, coffee, transportation, shopping, spa, internet, cleanliness, parking, gym, lunch, temperature, booking, restrooms
Beauty and spas (40, 352)	Staff, building, massages, service, hair, price, pedicure, location, bathrooms, products, shopping, spa, nails, pool, atmosphere, manicure, treatments, bedrooms, skin, food, facilities, face, amenities, bar, events, drinks, restaurant, coffee, booking, checks, cleanliness, dinner, lunch, breakfast, gym, parking, transportation, temperature, internet, restrooms
Restaurants (40, 361)	Food, service, menu, staff, vegetables, meat, sauces, potatoes, atmosphere, building, hamburgers, drinks, bread, food taste, price, seating, italian, location, dinner, desserts, cheese, bar, coffee, mexican, seafood, asian, rice, food quantity, lunch, breakfast, soups, appetizers, shopping, eggs, restrooms, parking, cleanliness, transportation, booking, temperature
Cell phones (19, 185)	Connectors, charger, protector, battery, appearance, earphones speaker, buttons, screen, price, sound, camera, connectivity, size, memory, weight, configuration, usage, microphone, processor
Movies and TV (23, 288)	Characters, cast, story, scenes, visual effects, script, music, picture, theme, locations, sounds, director, price, language, photography, start, costumes, visual style, writer, pacing, trailer, ending, atmosphere
Digital music (36, 381)	Song, album, singer, lyrics, sounds, music group, musician, guitar, rhythm, recording, music style, melody, theme, performance, start, harmony, instruments, drum, piano, timbre, picture, story, video, strings, price, characters, percussion, visual style, scenarios, texture, ending, cast, dynamics, trumpet, wind, costumes
CDs and vinyls (29, 260)	Song, album, singer, sounds, music group, lyrics, recording, musician, guitar, performance, music style, rhythm, theme, melody, harmony, start, drum, instruments, timbre, piano, video, price, strings, ending, percussion, texture, dynamics, wind, trumpet
Video games (20, 242)	Characters, controls, graphics, story, gameplay, scenarios, music, sounds, price, theme, script, configuration, customization, interface, difficulty, art style, start, pacing, ending, art style

For each vocabulary, the total numbers of aspects and nouns are given between parentheses, and the aspects are sorted in decreasing order of occurrences in the corresponding user review datasets

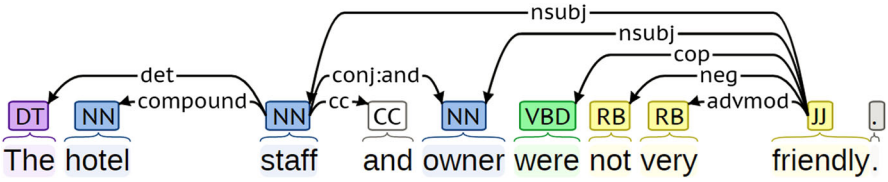

Fig. 1 POS tags and syntactic relations of the sentence "The hotel staff and owner were not very friendly"

50 grammatical relations. Figure 1 shows the POS tags and syntactic relations returned for the sentence "The hotel staff and owner were not very friendly".

The syntactic dependencies shown in the figure are given below as a list of triples. For instance, `(friendly-9, nsubj, staff-3)` means that the noun `staff` is the subject (`nsubj`) of the noun clause with the adjective `friendly`.

```
(ROOT-0, root, friendly-9)
(staff-3, det, The-1)
(staff-3, compound, hotel-2)
(friendly-9, nsubj, staff-3)
(staff-3, cc, and-4)
(staff-3, conj:and, owner-5)
(friendly-9, nsubj, owner-5)
(friendly-9, cop, were-6)
(friendly-9, neg, not-7)
(friendly-9, advmod, very-8)
```

For a given sentence, our method analyzes the list of syntactic dependencies to generate preliminary annotations in the form of `(noun, adjective, modifier, isAffirmative)` tuples, where `noun` and `adjective` are linked by certain syntactic relation (`nsubj` in general); `modifier`, if exists, is an adverb (e.g., 'little', 'enough', 'quite', 'very', 'absolutely') that may alter the polarity intensity of the adjective, to which is linked through the `advmod` relation; and `isAffirmative` is a Boolean variable that is 'true' if the polarity of the adjective has not to be inverted because there are not a `neg` relation or a 'but' preposition complementing the adjective, and the sentence is not negative.[8] In the previous example, the **voc** method would generate the following two tuples:

```
(staff, friendly, very, false)
(owner, friendly, very, false)
```

where 'staff' and 'owner' are noun *siblings* linked by the `conj:and` relation, and are described as 'very friendly', an adjective that, in this case, is not in an affirmative form since it is negated by the 'not' adverb. In Table 2 we show some examples of recognized sentence structures and generated opinion annotations, including affirmative versus negative sentences, single versus multiple nouns, single versus multiple adjectives, and adjective modifiers.

[8] Double negations of adjectives in sentences are also recognized by our method.

Table 2 Examples of aspect opinion tuples extracted by the **voc** method

Sentences	Aspect opinion tuples
"The hotel staff was friendly"	(staff, friendly, –, *true*)
"The staff of the hotel was friendly"	
"The hotel had a friendly staff"	
"I think the hotel staff was friendly"	
"The hotel had friendly and efficient staff"	(staff, friendly, –, *true*)
"The hotel had friendly, efficient staff"	(staff, efficient, –, *true*)
"The hotel staff and owner were friendly"	(staff, friendly, –, *true*)
'The hotel had friendly staff and owner"	(owner, friendly, –, *true*)
"The hotel staff and owner were friendly and efficient"	(staff, friendly, –, *true*)
'The hotel had friendly and efficient staff and owner"	(staff, efficient, –, *true*)
	(owner, friendly, –, *true*)
	(owner, efficient, –, *true*)
"The hotel staff was not friendly"	(staff, friendly, –, *false*)
"The staff of the hotel was not friendly"	
"The hotel had not a friendly staff"	
"The hotel had no friendly staff"	
"The hotel had non friendly staff"	
"I do not think the hotel staff was friendly"	
"The hotel staff was not friendly and efficient"	(staff, friendly, –, *false*)
"The hotel staff was neither friendly nor efficient"	(staff, efficient, –, *false*)
"The hotel staff and owner were not friendly"	(staff, friendly, –, *false*)
	(owner, friendly, –, *false*)

Table 2 continued

Sentences	Aspect opinion tuples
"The hotel staff and owner were not friendly and efficient"	(staff, friendly, –, *false*)
"The hotel staff and owner were neither friendly nor efficient"	(staff, efficient, –, *false*)
	(owner, friendly, –, *false*)
	(owner, efficient, –, *false*)
"The hotel had not a non friendly staff"	(staff, friendly, –, *true*)
"I do not think the hotel staff was not friendly"	
"The hotel staff was friendly, but not efficient"	(staff, friendly, –, *true*)
"The hotel staff was friendly, but was not efficient"	(staff, efficient, –, *false*)
"The hotel staff was friendly, but it was not efficient"	
"The hotel staff was not friendly but efficient"	(staff, friendly, –, *false*)
"The hotel staff was not friendly, but was efficient"	(staff, efficient, –, *true*)
"The hotel staff was not friendly, but it was efficient"	
"The hotel staff was very friendly"	(staff, friendly, very, *true*)

To generate the above annotations, A, we propose Algorithm 1, which processes certain syntactic patterns identified in a sentence S that relate nouns[9] and adjectives. Specifically, it analyzes the graph of dependencies D extracted by the CoreNLP tool (line 3), considering the following relations: `nsubj` and `nsubjpass`, which correspond to active/passive subjects in noun phrases—e.g., `(friendly, nsubj, staff)` in "The staff is friendly"—(lines 6–25), `amod` and `advmod`, which are adjectival and adverbial phrases complementing a noun phrase—e.g., `(staff, amod, friendly)` in "The hotel has friendly staff"—(lines 26–35), and `xcomp`, which represents predicative or clausal complements of a verb or adjective without its own subject—e.g., `(consider, xcomp, friendly)` in "I consider the staff friendly"—(lines 36–45). The algorithm analyzes other relations, such as `conj` and `xcomp` between pairs of nouns and pairs of adjectives/adverbs to extract noun siblings (function $getNounSiblings$ called in lines 7, 28 and 38) and adjective siblings (function $getAdjectiveSiblings$ called in lines 9, 17, 29 and 39) respectively, `acomp` and `advmod` to extract adjective modifiers (function $getAdjectiveModifiers$ called in lines 10, 18, 30 and 40), and `neg` to extract negations of adjectives (function $isAffirmative$ called in lines 11, 19, 31 and 41). Finally, the algorithm addresses the negation of the sentence by jointly considering the `root` and `neg` relations (lines 48-50), and removes those annotations whose nouns do not belong to the input, domain-dependent aspect vocabulary V (line 51).

As explained before, the proposed algorithm analyzes a sentence if it contains a noun that corresponds to an item aspect, i.e., a noun in the input vocabulary V. This does not allow extracting opinions about an aspect cited in a sentence through a personal pronoun (it, $they$), which refers to the aspect noun appearing in a previous sentence. To address this issue, we may use a coreference resolution technique. We tested the CoreNLP tool for such purpose, and decided to discard it in our experiments because the number of coreferences associated to aspects was very small, and the execution time increased significantly.

3.1.3 Opinion polarity identification

For each `(noun, adjective, modifier, isAffirmative)` annotation extracted by Algorithm 1, the **voc** method establishes the sentiment orientation of the opinion associated to the annotation, generating a final `(aspect, sentiment_orientation)` tuple, where `aspect` is the label of the aspect (in V) referred by `noun`, and `sentiment_ orientation` is a real number that is greater than, equal to, or lower than 0 if the opinion is positive, neutral or negative, respectively. In the following, we explain how such score is computed.

First, we set the adjective polarity $p_{adj} = polarity(\texttt{adjective}) \in \{-1, 0, +1\}$. We attempt to get such value from the well-known generic, domain-independent lexicon created by Hu and Liu (2004b). If the adjective is not found there, we attempt to obtain it from own domain-dependent, aspect-level lexicons, which we make publicly available[3]. We built the lexicon of a target domain extending the generic lexicon by

[9] The identification of nouns includes compound nouns, by means of the `compound`, `nn` and `nmod` relations.

```
input  : S- sentence; V- aspect vocabulary
output: A- list of aspect opinion annotations (noun, adjective, modifier, isAffirmative)
begin
    A ← list();
    D ← extractDependencies(S);
    for d ∈ D do
        switch d.dep do
            case nsubj, nsubjpass
                nouns ← getNounSiblings(D, d.dep);
                if isAdjective(d.gov) and not isVerbComplement(D, d.gov) then
                    adjs ← getAdjectiveSiblings(D, d.gov);
                    mods ← getAdjectiveModifiers(D, d.gov);
                    aff ← isAffirmative(D, d.dep);
                    a ← annotations(d.dep, d.gov, nouns, adjs, mods, aff);
                    A.add(a);
                else if isVerb(d.gov) then
                    for d' ∈ D do
                        if d'.rel = xcomp and d'.gov = d.gov then
                            adjs ← getAdjectiveSiblings(D, d'.dep);
                            mods ← getAdjectiveModifiers(D, d'.dep);
                            aff ← isAffirmative(D, d.dep);
                            a ← annotations(d.dep, d'.dep, nouns, adjs, mods, aff);
                            A.add(a);
                        end
                    end
                end
            end
            case amod, advmod
                if isAdjective(d.dep) or isVerbGerund(d.dep) or
                (isAdverb(d.dep) and (isNoun(d.gov) or isPronoun(d.gov))) then
                    nouns ← getNounSiblings(D, d.gov);
                    adjs ← getAdjectiveSiblings(D, d.dep);
                    mods ← getAdjectiveModifiers(D, d.dep);
                    aff ← isAffirmative(D, d.gov);
                    a ← annotations(d.gov, d.dep, nouns, adjs, mods, aff);
                    A.add(a);
                end
            end
            case xcomp
                if (isNoun(d.dep) or (isPronoun(d.dep) or
                isVerbGerund(d.dep))) and isAdjective(d.gov) then
                    nouns ← getNounSiblings(D, d.dep);
                    adjs ← getAdjectiveSiblings(D, d.gov);
                    mods ← getAdjectiveModifiers(D, d.gov);
                    aff ← isAffirmative(D, d.dep);
                    a ← annotations(d.dep, d.gov, nouns, adjs, mods, aff);
                    A.add(a);
                end
            end
        endsw
    end
    if isNegativeSentence(S) then
        A.invertPolarities();
    end
    A.removeNonAspectAnnotations(V);
end
```

Algorithm 1: Extraction of aspect opinion annotations from a sentence.

computing $PMI(a_g, a_d)$ values (see Sect. 2.1.2) between adjectives a_g and a_d that co-occur in aspect opinions of a review collection on the domain, where a_g is an adjective of the generic lexicon, and a_d is an adjective whose polarity is unknown. For those pairs that have PMI values greater than certain threshold (chosen by manual inspection), the polarity of a_g determines the polarity of a_d. Thus, for example, if "expensive" and "small" appear frequently together when describing the *size* of rooms in hotel reviews, they would have a high PMI value, and, since the polarity of "expensive" is negative, we set the polarity of "small" to negative (for the hotel *room size* aspect).

Next, if exist, we consider adverbs to strengthen or soften the `adjective` polarity. This is a particular case of the intensifiers discussed by Taboada et al. (2011), and is envisioned as an open research issue in Chen et al. (2015). Our method makes use of a list of 300 adverbs, each of them with a weight $w_{mod} \in \{-1, +0.5, +2\}$ expressing respectively whether the adverb inverts, softens or strengthens the polarity of the adjective. If the `modifier` of the annotation belongs to that list, we set the corresponding weight w_{mod}. The list, which we also make publicly available, is composed of the Thesaurus.com[10] synonyms of representative adverbs, namely *very*, *entirely*, *amazingly*, *quite*, *somehow*, *little*, *too*, *excessively* and *insufficiently*, and the synonyms of the latter, discarding duplicates. More specifically, the list contains 83, 82 and 135 adverbs with weights $w_{mod} = -1, +0.5, +2$, respectively.

Finally, we take the `isAffirmative` value into account to set a weight $w_{aff} = \{-1, +1\}$ depending on whether `isAffirmative` is $false$ or $true$, respectively. The value of `sentiment_orientation` is then computed as follows:

$$\texttt{sentiment\_orientation} = w_{aff} \cdot w_{mod} \cdot p_{adj} \in \{-2, -1, -0.5, 0, +0.5, +1, +2\}$$

As illustrative examples, "amazingly tasty" and "slightly expensive" are assigned $+2$ and -0.5 semantic orientation values, respectively.

3.2 SABRE method

As a representative method of word frequency-based aspect opinion extraction approaches, we have implemented the SABRE algorithm (Caputo et al. 2017). Making use of Language Models, this algorithm works on the assumption that the vocabulary used differs when talking about distinct topics. Hence, it aims at selecting as aspects the nouns whose distributions in a specific-domain document collection differs significantly from their distributions in a general, multi-domain corpus. Caputo et al. (2017) conducted experiments over a set of TripAdvisor[11] reviews, showing that using their KL divergence metric allowed extracting better aspects than considering only frequencies of appearance. We will refer to this method as **sab** in the remainder of the document.

[10] Thesaurus.com - synonyms and antonyms, http://www.thesaurus.com.

[11] TripAdvisor travel and restaurant review site, https://www.tripadvisor.com.

3.2.1 Aspect extraction

Assuming that aspects are mostly nouns (Liu 2012), we compute the frequency of appearance of each noun in the specific item domain. Similarly to Caputo et al. (2017), we utilize the Stanford CoreNLP *lemmatizer* to consider two nouns to be the same if they have a common lemma. Formally, we compute the frequency and subsequent probability of lemma t appearing $n_{t,D}$ times in domain D as

$$p_{t,D} = p(t, D) = \frac{n_{t,D}}{N_D}$$

where N_D is the sum of the frequencies of all the noun lemmas in the domain.

Next, $p_{t,D}$ is compared to the probability of appearance of t in a general, multi-domain corpus. As done in Caputo et al. (2017), we use the British National Corpus (BNC)[12] to do this comparison. The method assigns a score to every noun in the domain that also appears in the generic corpus as the pointwise Kullback–Leibler divergence δ between both probabilities. Let D be the target domain, and BNC be the multi-domain corpus, the above score is calculated as

$$score(t) = \delta_t(D||BNC) = p(t, D) \log \frac{p(t, D)}{p(t, BNC)}$$

Finally, every noun with a score higher than a threshold ε is considered to be an aspect, since it is overrepresented in the target domain. The authors set $\varepsilon = 0.3$ in reported experiments.

3.2.2 Opinion polarity identification

Differently to the method proposed in Caputo et al. (2017), we follow the algorithm explained in Sect. 3.1.3 to identify the sentiment orientation of the existing opinions about the extracted aspects. We refer the reader to that section for the details. We just remind the reader here that our algorithm allows considering both adjective and sentence negation, adjective modifiers, and multiple aspects and opinions in a sentence.

3.3 Double propagation method

In our experiments, we also consider a syntactic relation-based method to aspect opinion extraction. In particular, we evaluate the Double Propagation (DP) method presented in Qiu et al. (2011). The DP algorithm has become the basis of several state-of-the-art methods for identifying opinions about item aspects from textual reviews. This method is based on the observation that aspects are mostly nouns, and opinion words are mostly adjectives complementing such nouns. Hence, analyzing (noun, adjective) syntactic relations, the **dp** method aims at finding aspect opinions and their sentiment orientations simultaneously.

[12] British National Corpus, http://www.natcorp.ox.ac.uk.

3.3.1 Aspect extraction

The basic idea of the **dp** method is to identify aspect and opinion words iteratively using known and extracted (in previous iterations) aspect and opinion words, and certain syntactic relations, propagating information back and forth between iterations.

The identification of the relations is the key to the extraction. Two words are *direct dependent* if one word depends on the other word without any additional words in their grammar dependency path, or if both have a direct dependency on a third word. In particular, **dp** uses direct dependencies between nouns and adjectives, identified by the POS tags: *NN* (nouns) and *NNS* (plural nouns) for aspects, and *JJ* (adjectives), *JJR* (comparative adjectives) and *JJS* (superlative adjectives) for opinions. As done by Qiu et al. (2011), we obtained these tags with the Stanford POS tagger (Socher et al. 2013).

The *mod, pnmod, subj, s, obj, obj2, desc* syntactic relations were considered between an aspect word and an opinion word, whereas the *conj* relation was used between aspect (or opinion) words. The followed procedure to find such syntactic dependencies in the reviews is similar to that exposed in Sect. 3.1.2. We run the POS tagger and syntactic dependencies parser to obtain triples of the form (`gov, rel, dep`) that represent the relations `rel` hold between governors `gov` and dependents `dep`; see Sect. 3.1.2 for details. In the following, for simplicity, we assume (`gov, rel, dep`) and (`rel, gov, dep`) are equivalent, and use (`word1, word2, dep`) to refer to both of them. The developed method handles both alternatives.

After the nouns, adjectives and dependency relations are identified, the propagation algorithm starts. The four rules proposed in Qiu et al. (2011) are presented in Table 3. They are used to extract new words from previously extracted words. The **dp** method begins with a list of well-known opinion words from the Lexicon of Liu et al. (2012). In the first iteration, considering the initial words, the method extracts related aspect words through Rule 1, and other opinion words through Rule 4. Then, it searches for nouns or opinion words related to these new extracted words through Rules 2 and 3, parsing every sentence in the dataset. The procedure is repeated until no more aspect or opinion words are extracted following the propagation.

When the propagation has finished, we run a pruning stage to remove noise terms that have been selected as potential aspects. We perform a modified version of the *Clause Pruning* suggested in Qiu et al. (2011), which consists in keeping only the most frequent target noun in a clause with several nouns. In terms of Precision, Recall and F-score in aspect extraction, we compared the results obtained with *Clause Pruning* against *Sentence Pruning*—i.e., keeping only the most popular word in a sentence as target aspect—on the dataset used by Liu (2012). We did not observe significant differences in performance, but *Sentence Pruning* avoids parsing the sentence to obtain the clauses, and is more scalable. Therefore, in this work we apply *Sentence Pruning* instead of *Clause Pruning*. We also perform a global pruning stage, removing target words that appear only once in the whole opinion set. Finally, we perform *Compound Pruning*, which combines multiple words (two nouns or a noun and an adjective) to create multi-term aspects. We will refer to the DP + pruning method as **dpp**.

Table 3 Double propagation extraction rules

Rule	Relations	Known word	Extracted word	Example sentences	Relation
R1	(*JJ*, *NN*, MR)	*JJ*	*NN*	The phone has a good "screen"	(*good*, *screen*, *mod*)
	(*JJ*, H, MR)	*JJ*	*NN*	"iPod" is the best mp3 player	(*best*, *player*, *mod*)
	(H, *NN*, MR)				(*player*, *iPod*, *subj*)
R2	(*JJ*, *NN*, MR)	*NN*	*JJ*	Similar to R1, but the noun as the	
	(*JJ*, H, MR)	*NN*	*JJ*	known word and the adjective as	
	(H, *NN*, MR)			the extracted word	
R3	(*NN*, *NN*, CONJ)	*NN*	*NN*	Does the player play dvd with	(*video*, *audio*, *conj*)
	(*NN*, H, T)	*NN*	*NN*	audio and "video"?	(*lens*, *has*, *obj*)
	(H, *NN*, T)			Canon "G3" has a great lens	(*has*, *G3*, *subj*)
R4	(*JJ*, *JJ*, CONJ)	*JJ*	*JJ*	The camera is amazing and "easy"	(*easy*, *amazing*, *conj*)
	(*JJ*, H, T)	*JJ*	*JJ*	to use	(*sexy*, *player*, *mod*)
	(H, *JJ*, T)			If you want to buy a sexy, "cool"	(*player*, *cool*, *mod*)
				accessory-available mp3 player,	
				you can choose iPod	

Underlined words are the known words, and the words with double quotes correspond to extracted words. MR represents the relations *mod, pnmod, subj, s, obj, obj2, desc*, and CONJ corresponds to the relation *conj*. H refers to any word appearing in both dependencies, and T represents a dependency on the same family

3.3.2 Opinion polarity identification

In the **dp** method, the assignment of polarity to adjectives is done simultaneously to the propagation process. The polarity of a new extracted word depends on the polarity of the word from which it has been propagated. The underlying idea is that the syntactic relations used in the extraction rules correspond to dependencies that refer to the same concept, so they have to share the polarity.

The initial words are annotated with the polarity scores (+1 for positive, -1 for negative) existing in a lexicon, and these scores are then used in the propagation. Moreover, the assigned polarity scores are inverted if negation words affect the extracted words, within a surrounding word window. In our experiments, we set a 5-word window as in the original work. We refer to Qiu et al. (2011) for more details on this sentiment orientation assignment.

3.4 LDA method

Topic model-based aspect opinion extraction methods provide latent representations of items (and users) in terms of the topics discussed in the reviews. These representations allow extracting intrinsic characteristics of the items from their reviews, which capture, among other things, the item aspects commented by the users. In particular, we evaluate the standard form of LDA, an effective algorithm that is the basis of the state-of-the-art methods based on topic models. We will refer to this method as **lda**.

3.4.1 Aspect-topic representation

As discussed in Sect. 2.1.4, LDA may extract generic topics, instead of specific topics related to aspects. In this context, we have empirically observed that as the number of topics increases, the obtained latent topics are more aspect-specific. In fact, we will report results of experiments run on up to 100 topics.

We use the LDA implementation of the MALLET framework (McCallum 2002), optimizing the hyperparameters every 20 iterations. We run the algorithm for at least 500 iterations, until convergence ($\varepsilon = 0.001$) in the logarithm of the perplexity metric. As done by McAuley and Leskovec (2013), we consider the set of all reviews of a particular item as a document, which leads to the latent representation of the item.

3.4.2 Opinion polarity identification

LDA allows representing an item as a K-dimensional vector $(\varphi_{i,1}, \varphi_{i,2}, \ldots, \varphi_{i,K})$, where $\varphi_{i,k}$ is the proportion of item **i** about topic (aspect) a_k. The assigned polarity $w_{i,k}$ of aspect a_k to item **i** is computed as the weighted sentiment orientation $so_{i,k}$ of the topic k in item **i**:

$$w_{i,k} = \varphi_{i,k} \cdot so_{i,k}$$

where $so_{i,k}$ is computed by selecting the 10 most representative words for topic k, and computing the average polarity of those words in the document (i.e., the set of

reviews about item **i**). The polarity of these words is computed following the algorithm explained in Sect. 3.1.3.

4 Developed aspect-based recommendation methods

To analyze the effect of exploiting opinions about item aspects in recommender systems, we experiment with two families of recommendation approaches: content-based and collaborative filtering.

The opinion information to be used by the evaluated recommenders will be generated by the aspect extraction methods presented in Sect. 3, namely the vocabulary-based **voc** method, the word frequency-based **sab** method, the syntactic relation-based **dp** and **dpp** methods, and the latent topic-based **lda** method.

Depending on the particular combinations of aspect extraction and recommendation approaches, the recommenders will belong to one of the types of aspect-based recommendation presented in Sect. 2.2: building enhanced aspect-based item profiles (Sect. 2.2.1), modeling latent user preferences on aspects (Sect. 2.2.2), setting the weights of aspect-level user preferences (Sect. 2.2.3), and incorporating aspect-based user/item similarities into recommendation heuristics (Sect. 2.2.4).

Before presenting in Sect. 4.2 the particular evaluated recommenders, in Sect. 4.1 we first explain how user and item profiles are built with aspect-based information extracted from reviews.

4.1 Modeling users and items

Following the standard procedure proposed in the literature (Chen et al. 2015), we split the aspect-based modeling process into item profiling (Sect. 4.1.1) and user profiling (Sect. 4.1.2).

From now on, a user u_m's profile is represented as a vector $\mathbf{u}_m = \{w_{m,1}, w_{m,2}, \ldots, w_{m,K}\}$, where $w_{m,k}$ denotes the relative relevance (weight) of aspect a_k for u_m, and K is the total number of aspects. Analogously, an item i_n's profile is represented as a vector $\mathbf{i}_n = \{w_{n,1}, w_{n,2}, \ldots, w_{n,K}\}$, where $w_{n,k}$ denotes the relative relevance of a_k for i_n.

4.1.1 Item profiling

Next, we describe how we compute the weight $w_{i,k}$ for each item i and aspect a_k. We consider both profiles associated to actual aspects commented in the user reviews, and profiles composed of latent (aspect) topics inferred from the texts of the review collection.

Aspect annotation-based item profiles This type of item profiling assumes that the aspect extraction technique may generate tuples $(u, i, a_k, so_{u,i,k})$ for each user u and item i, associated to aspect a_k and sentiment orientation $so_{u,i,k}$. In this case, the weight of an aspect for a particular item is computed as the average of the estimated sentiment orientation over every occurrence such aspect appears in the

reviews associated to that item. Formally:

$$w_{i,k} = \frac{1}{|(\cdot, i, a_k, \cdot)|} \sum_{(\cdot,i,a_k,so)} so_{u,i,k} \tag{1}$$

Latent factor-based item profiles This profiling technique is used together with latent topic-based aspect extraction methods, which represent the items by their topic distribution. As described in Sect. 3.4.2, a sentiment orientation $so_{i,k}$ can be assigned to each aspect (latent topic) for every item. Furthermore, an item i can be represented in terms of the proportion $\varphi_{i,k}$ of each aspect k, leading to the following weight for each (item, aspect) pair:

$$w_{i,k} = \varphi_{i,k} \cdot so_{i,k} \tag{2}$$

4.1.2 User profiling

The user profiles are defined in the same vector space as the item profiles presented in the previous subsection. They, however, are built with two different strategies: aggregating explicit user opinions about aspects, and implicitly aggregating such information from aspect-based item profiles. In this context, we do not consider explicit latent factor-based user profiles, where co-clustering techniques would be needed for generating topic models (Kumar et al. 2016), since classical topic models build topics based on a specific dimension, either users or items.

Explicit aspect-based user profiles In the simplest form, the weight of an aspect for a particular user is computed as the average of the estimated sentiment orientation over every explicit occurrence such aspect appears in the reviews written by that user, using the tuples $(u, i, a_k, so_{u,i,k})$ generated by any aspect extraction technique. Formally:

$$w_{u,k} = \frac{1}{|(u, \cdot, a_k, \cdot)|} \sum_{(u,\cdot,a_k,so)} so_{u,i,k} \tag{3}$$

We shall denote the recommenders using this profile type with the term **exp**.

Implicit aspect-based user profiles In this case, a user's profile is generated with the profiles of those items rated by the user. More specifically, the user's profile is the aggregation of the aspect-based profiles of the items reviewed (thus, rated) by the user, weighted by the user's ratings. Hence, the user's preferences for item aspects are implicitly inferred. Formally, the weight $w_{u,k}$ of an aspect a_k for a particular user u is computed as follows:

$$w_{u,k} = \sum_{\{(u,i,r),r\neq\emptyset\}} r(u, i) \cdot w_{i,k} \tag{4}$$

We shall denote the recommenders using this profile type with the term **imp**.

4.2 Aspect-based recommendation methods

Once the profiles for users and items are generated, they are exploited by content-based and collaborative filtering methods to provide personalized recommendations. In the following subsections, we present the formulations defined for each method to estimate an unknown rating $\hat{r}(u_m, i_n)$.

4.2.1 Content-based methods

A pure content-based method only relies on the aspect-based representations of users and items, without exploiting rating data. In our experiments, we evaluate the **cb** method, which returns the cosine similarity between the above representations, that is:

$$\hat{r}(u_m, i_n) = \cos(\mathbf{u}_m, \mathbf{i}_n) = \frac{\sum_{k=1}^{K} w_{m,k} \cdot w_{n,k}}{\sqrt{\sum_{k=1}^{K} (w_{m,k})^2 \sum_{k=1}^{K} (w_{n,k})^2}} \tag{5}$$

We will refer to this method with different names, depending on the item aspect opinion extraction and user profile used. Specifically, we shall follow the notation **cb-*asp*-*up***, where *asp* refers to a particular aspect opinion extraction method, and *up* denotes certain user profiling technique. For instance, producing recommendations with **cb-lda-imp** would mean that item aspect opinions were extracted by the **lda** method, and the user profiles were built with the **imp** technique, i.e., using Eq. 4. Hence, all the instances of the method are **cb-*asp*-exp** and **cb-*asp*-imp**, where *asp* can be **voc**, **sab**, **dp** or **dpp**.

4.2.2 Collaborative-via-content hybrid methods

Collaborative patterns in aspect opinion data can be exploited by adapting a nearest neighbor CF algorithm, so that a content-based user/item similarity is used instead of a pure rating-based similarity. This is actually the idea behind the *collaborative-via-content* hybrid recommendation method proposed in Pazzani (1999), which has been shown to achieve good performance results, as it combines the advantages of both content-based and collaborative filtering.

In particular, we perform two variations of such a hybrid recommender: one based on item similarities (Eq. 6) and another based on user similarities (Eq. 8). Both algorithms are inspired by the classical nearest neighbor CF heuristics: item- and user-based nearest neighbors, respectively (Ning et al. 2015). In particular, we use some recent formulations optimized for ranking, where the similarity normalization factor is removed (Cremonesi et al. 2010).

The item-based hybrid method, **cbib**, is formulated as:

$$\hat{r}(u_m, i_n) = \sum_{j \in N_l(i_n)} sim(i_n, j) \cdot r(u_m, j) \tag{6}$$

where $N_l(i_n)$ denotes the l items most similar to i_n and $sim(\cdot, \cdot)$ is a content-based item similarity metric based on the corresponding item profiles, such as the cosine similarity:

$$sim(i_n, i_j) = \cos(\mathbf{i}_n, \mathbf{i}_j) = \frac{\sum_{k=1}^{K} w_{j,k} \cdot w_{n,k}}{\sqrt{\sum_{k=1}^{K} (w_{j,k})^2 \sum_{k=1}^{K} (w_{n,k})^2}} \tag{7}$$

On the other hand, the user-based hybrid method, **cbub**, is formulated as:

$$\hat{r}(u_m, i_n) = \sum_{v \in N_l(u_m)} sim(u_m, v) r(v, i_n) \tag{8}$$

where $N_l(u_m)$ denotes the l users most similar to u_m and $sim(\cdot, \cdot)$ is a content-based user similarity metric based on the corresponding user profiles, such as the Cosine similarity computed as in Eq. 7.

Following the notation of the content-based methods, we also include the types of user and item profiles into the names of the collaborative filtering (hybrid) methods. Since LDA does not allow for explicit user profiles, we only consider the **cbib-lda-imp** recommender, which implements Eq. 6, and the **cbub-lda-imp** recommender, which implements Eq. 8. Additionally, for the rest of the aspect extraction methods, and taking into account that **cbib** may not exploit the user profiles, we could generate recommendations using any of the three following combinations: **cbib-*asp***, **cbub-*asp*-exp**, and **cbub-*asp*-imp**, where *asp* is either **voc**, **sab**, **dp** or **dpp**.

5 Experimental setting

Next, we describe some issues about our experiments, namely the used datasets (Sect. 5.1), the followed evaluation methodology and analyzed metrics (Sect. 5.2), and the evaluated aspect opinion extraction and recommendation methods (Sects. 5.3 and 5.4).

5.1 Datasets

In order to provide well argued conclusions about the effectiveness of exploiting item aspect opinions by recommender systems, we have evaluated the developed methods on several domains. As done by other researchers (see Sect. 2), we used two popular datasets, namely the Yelp challenge[1] and the Amazon product reviews[2] (McAuley and Yang 2016) datasets. From the Yelp dataset, we used all its reviews about *Hotels* (HOT), *Beauty and Spas* (SPA) and *Restaurants* (RES), which do have a relatively large number of user opinions about item aspects. From the Amazon dataset, we selected the reviews about *movies* and *music*—specifically *Movies and TV* (MOV), *Digital Music* (MUS) and *CDs and Vinyls* (CDS)—since historically they have been the most popular application domains in the recommender systems field, and *Cell phones* (PHO) and *Video Games* (GAM) since they contain items whose aspects are very frequently reviewed on the Web. Statistics about these datasets are shown in Table 4. They cover ranges from a few thousands to more than one and a half million reviews. As can be seen in the table, the Yelp datasets do have relatively few ratings per

Table 4 Summary of statistics about the datasets used in the experiments

Dataset	Domain	Abb	Ratings	Users	Items	Rating density	Ratings/user	Ratings/item
Yelp	Hotels	HOT	5034	4148	284	$4.27 \cdot 10^{-3}$	1.21	17.73
	Beauty and spas	SPA	5579	4272	764	$1.71 \cdot 10^{-3}$	1.31	7.3
	Restaurants	RES	158,430	36,473	4503	$9.65 \cdot 10^{-4}$	4.34	35.18
Amazon	Movies and TV	MOV	1,697,533	123,960	50,052	$2.74 \cdot 10^{-4}$	13.69	33.92
	Digital music	MUS	64,706	5541	3568	$3.27 \cdot 10^{-3}$	11.68	18.14
	CDs and vinyls	CDS	1,097,592	75,258	64,443	$2.26 \cdot 10^{-4}$	14.58	17.03
	Cell phones	PHO	194,439	27,879	10,429	$6.69 \cdot 10^{-4}$	6.97	18.64
	Video games	GAM	231,780	24,303	10,672	$8.94 \cdot 10^{-4}$	9.54	21.72

user in comparison to the Amazon datasets, which may be in detriment of collaborative filtering methods.

5.2 Evaluation methodology and metrics

In the experiments, we performed 5-fold cross validation to split a dataset into 5 training and 5 test subsets with which computing average recommendation performance results. Since in the recommender systems field rating prediction metrics, such as MAE and RMSE, are progressively in disuse, we focused our evaluation on ranking-based metrics. For such purpose, we generated the recommendation rankings following the *TrainingItems* methodology described in Bellogín et al. (2011), where every item in the training data split, except those known (rated/reviewed) by the target user, is considered as a possible candidate for the user's final recommendation list.

The reported metrics (using the implementation provided in the RankSys framework[13]) are the following:

- **P** (precision) and **R** (recall): these metrics measure the amount of relevant returned items, either normalized by the amount of items returned (precision) or the amount of relevant items known for each user (recall).
- **nDCG** (normalized discounted cumulative gain): this metric allows considering differences in the ranking positions of the relevant returned items, positively scoring relevant items recommended in the first positions of the rankings (Bellogín et al. 2011).
- **USC** (user space coverage): this metric allows considering the tradeoff between recommendation quality (as measured by the previous metrics) and the amount of users who receive recommendations (user coverage).
- **AD** [aggregate diversity (Castells et al. 2015)]: this metric measures the number of different items a recommender is able to provide. It is thus related to recommendation diversity, since the larger that number, the more diverse the recommendation lists presented to the users.

[13] RankSys recommender systems evaluation framework, http://ranksys.org.

– **EPC** [expected popularity complement (Castells et al. 2015)]: this metric measures the expected number of relevant items not previously seen. It is thus related to recommendation novelty.

For these metrics, we tested several cutoffs, but decided to report the performance at 5 to emphasize performance at top positions of the recommendation lists.

5.3 Evaluated aspect extraction methods

As presented in Sect. 3, covering the approach types existing in the literature, we have evaluated the next aspect extraction methods:

– **voc** The vocabulary-based method that exploits manually chosen aspect terms, as those given in Table 1.
– **sab** SABRE, the frequency-based method that selects terms that have a high ratio of appearance in the target domain with respect to their appearance in a general, multi-domain corpus. We selected aspects with a score higher than a threshold ε, for $\varepsilon = \{0.1, 0.05, 0.03, 0.01, 0.005, 0.003, 0.001\}$.
– **dp** Double Propagation, the syntactic dependency-based method that selects aspect terms based on syntactic relations between nouns and adjectives in sentences. We selected the top $N = 10, 20, 50, 100, 200$ and 500 most frequent terms as aspects.
– **dpp** The Double Propagation method with a subsequent pruning stage.
– **lda** LDA, the topic model-based method that represents items in terms of the topics discussed in their reviews. We generated 5, 10, 20, 50 and 100 aspects (topics).

5.4 Evaluated recommendation methods

We have evaluated a number of recommendation methods, implemented on top of the RankSys framework for replicability purposes. Specifically, we have evaluated two baseline methods that provide non personalized recommendations:

– **rnd** A recommender that generates random scores for each user-item pair.
– **ipop** An item popularity-based recommender that recommends to all users the items with more ratings, without considering any personal preferences.

We have also evaluated standard content-based and collaborative filtering methods as non aspect-based baselines:

– **cb** The content-based recommendation method that exploits the user and item profiles presented in Sect. 4.1. The score produced by this method is the cosine similarity between the user's profile and the profile of every candidate item (not previously seen by the user) in the system, as presented in Sect. 4.2.1.
– **ib** An item-based nearest neighbor method that exploits the rating-based similarity between items to create neighborhoods, which are used to compute a score for each (user, item) pair. We used the cosine similarity without any constraint on the neighborhood size; hence, the neighborhood is limited to the items rated by the target user.

- **ub** A user-based nearest neighbor method that works in a similar way as **ib**, but computing the similarities between users. We used the cosine similarity, and tested several neighborhood sizes, namely $l = 5, \ldots, 100$, in steps of 5.
- **mf** A matrix factorization collaborative filtering method. We used the variation proposed in Hu et al. (2008) (the HKV factorizer implemented in RankSys), since it has shown very good performance in different datasets. We tested several numbers of latent factors: from 5 to 100, in steps of 5.

Moreover, he have evaluated collaborative-via-content hybrid recommenders, which apply collaborative filtering heuristics using content-based user/item similarities. More specifically, such similarities are computed with aspect opinion information as explained in Sect. 4.2.2:

- **cbib** A hybrid recommendation method where an item-based CF heuristic is computed using content-based item similarities. More specifically, it computes a cosine similarity in a similar way as in **cb**, but between two item profiles instead of between a user and an item profiles; then, it uses the standard formulation followed by **ib**. In the experiments, we tested several values for the neighborhood size l: from 5 to 100, in steps of 5.
- **cbub** A hybrid recommendation method where a user-based CF heuristic is computed using content-based user similarities. It follows the same development as in **cbib**, but computing similarities between two user profiles instead of item profiles. As done with **cbib**, we tested different values for l: from 5 to 100, in steps of 5.

Finally, we have evaluated a state-of-the-art aspect-based recommender:

- **hft** The HFT algorithm (McAuley and Leskovec 2013), which builds a matrix factorization model that incorporates hidden topics as a proxy for item aspects (see Sect. 2.2.2).

6 Evaluation results

In this section, we present the experiments conducted to address our research questions, namely RQ1, *is there an aspect extraction method that generates data consistently effective for both content-based and collaborative filtering strategies?*, RQ2, *to what extent are opinions about item aspects valuable to improve the quality of personalized recommendations?*, and RQ3, *how do the type and coverage of extracted aspects affect the performance of aspect-based recommendation methods?*

The analysis of the achieved empirical results is structured as follows. In Sects. 6.1 and 6.2 we discuss main conclusions regarding the accuracy, novelty, diversity and novelty of recommendations generated by the developed methods. Next, in Sect. 6.3, we study alternative scenarios with respect to the quality/quantity of aspect opinion annotations in input reviews, and analyze the impact that the item catalog coverage of the aspect extraction methods has on subsequent recommendations.

Table 5 Comparison of aspect-based recommenders performance values (measured as P@5) on each domain

rec	asp	up	YELP			AMAZON				
			HOT	SPA	RES	MOV	MUS	CDS	PHO	GAM
cb	voc	imp	0.017	0.005	0.005	0.001	0.001	0.000	0.001	0.001
cb	voc	exp	0.008	0.008	0.001	0.000	0.003	0.000	0.001	0.001
cb	sab	imp	**0.027**	0.003	**0.007**	0.000	0.005	0.000	0.002	0.002
cb	sab	exp	0.007	0.003	0.003	0.000	0.002	0.000	0.000	0.000
cb	dp	imp	0.018	0.008	0.004	0.001	0.005	0.001	0.002	0.002
cb	dp	exp	0.010	0.004	0.002	0.000	0.003	0.000	0.000	0.001
cb	dpp	imp	0.014	**0.009**	0.005	0.002	0.007	0.002	0.002	0.002
cb	dpp	exp	0.009	0.004	0.002	0.000	0.002	0.001	0.000	0.001
cb	lda	imp	0.020	0.007	0.003	**0.006**	**0.021**	**0.009**	**0.002**	**0.005**
cbib	voc	–	0.019	0.005	0.004	0.001	0.001	0.000	0.000	0.001
cbib	sab	–	**0.029**	0.005	**0.007**	0.000	0.005	0.000	0.002	0.002
cbib	dp	–	0.019	0.007	0.004	0.001	0.005	0.001	0.001	0.002
cbib	dpp	–	0.014	0.008	0.005	0.002	0.007	0.002	0.001	0.002
cbib	lda	–	0.021	**0.009**	0.005	**0.015**†	**0.042**	**0.017**	**0.005**	**0.010**
cbub	voc	imp	0.022	0.007	0.010	0.006	0.020	0.006	0.005	0.005
cbub	voc	exp	**0.035**†	**0.023**†	0.007	0.001	0.006	0.001	0.003	0.002
cbub	sab	imp	0.029	0.010	0.012	0.000	0.029	0.000	0.010	0.011
cbub	sab	exp	0.028	0.009	0.009	0.000	0.008	0.000	0.003	0.003
cbub	dp	imp	0.026	0.013	0.012	0.011	0.033	0.013	0.011	0.016
cbub	dp	exp	0.027	0.010	0.008	0.003	0.019	0.004	0.007	0.007
cbub	dpp	imp	0.019	0.013	0.012	0.012	0.033	0.013	0.011	0.016
cbub	dpp	exp	0.023	0.010	0.008	0.004	0.020	0.004	0.006	0.008
cbub	lda	imp	0.028	0.014	**0.013**†	**0.015**	**0.046**†	**0.020**†	**0.012**†	**0.018**†

asp and up denote the corresponding item aspect extraction and user profiling techniques. The best method combination for each domain and recommender is in bold and for each domain is marked with †

6.1 Analyzing recommendation quality: accuracy-based evaluation

On each of the considered domains, and in terms of P@5, Table 5 shows the best performance achieved by every combination of recommendation (rec) and item aspect extraction (asp) methods and user profiling (up) technique, according to what was presented in Sect. 4. We omit the performance results with recall and nDCG metrics, since they behave similarly to precision. Moreover, for the sake of reproducibility, in Table 9 at the "Appendix" of this paper we present the values of the input parameters of all the tested methods.

According to the achieved results (where all the differences are statistically significant, using the RiVal toolkit's[14] implementation of the Wilcoxon paired test for $p < 0.05$), we could distinguish between three groups of domains. A first group would

[14] RiVal recommender system evaluation toolkit, http://rival.recommenders.net.

be composed of the Hotels (HOT) and Beauty and spas (SPA) domains from the Yelp datasets. For these domains, the optimal recommender is the hybrid **cbub** method using **voc** for aspect opinion extraction and **exp** as aspect-based user profiling technique. There is certain gap in the precision values of **cbub** with the remainder aspect extraction methods (**sab**, **dp**, **dpp** and **lda**), but also in these cases the recommender achieves better precision than **cbib** and **cb** in general. Depending on the recommendation method, it is better to use an explicit representation of the user profiles (for **cbub**) or, conversely, an implicit representation (for **cb**); nonetheless, the best results are achieved with the explicit ones. The HOT and SPA domains do have reviews with abundant aspect opinions, since the Yelp system is devoted to allow users to upload and vote for personal reviews. Moreover, it was quite straightforward for the **voc** method to manually identify the relevant item aspects of these domains (see Table 1); in fact, as shown in Table 6, **voc** obtained high annotation coverage: 97.4% and 81.0% of the available user reviews for HOT and SPA.

A second group would be associated to the Restaurants (RES) domain from the Yelp dataset, for which the **cbub** hybrid recommender again achieves the best precision values, but where there are no clear differences on performance when using **sab**, **dp** or **dpp**, and using **lda**, the best performing aspect extraction method. Moreover, in contrast to the first group of domains (HOT and SPA), on the RES domain the implicit representation of user profiles is the best choice in all cases. As for HOT and SPA, the RES dataset comes from the Yelp system, and thus has many aspect opinions; in fact, the **voc** method also annotated most of the available user reviews, 96.9% (see Table 6). However, as commented in Sect. 3.1.1, on RES we considered certain aspects, mainly related to principal ingredients of the restaurants dishes, which represent general cuisine topics instead of particular aspects of the reviewed restaurants. These topics are highly discussed in the reviews, which benefits the item semantic clustering made by **lda**.

Finally, in a third group, we would have all the domains of the Amazon datasets: Movies (MOV), Digital music (MUS), CDs and vinyls (CDS), Cell phones (PHO), and Video games (GAM). On these domains, once more, the **cbub** hybrid method outperforms **cbib** (except for MOV where the differences are not significant), and the pure content-based **cb** method is the worst performing one. Differently to the first group, regarding the aspect extraction, **lda** achieves the best precision values, followed by **dpp**, **dp**, **sab**, and lastly **voc**. Additionally, for the **cbub** and **cb** methods, we observe that the implicit aspect-based user profiling **imp** outperforms again its explicit counterpart. The datasets of these domains come from the Amazon e-commerce site, which is not focused on user reviews; the coverage of **voc** method was approximately 50% on average for all domains except MUS (see Table 6). For this reason, it is not surprising that the **lda** method outperforms the other aspect extraction methods, which aim to extract explicit references to item aspects from user reviews.

Summarizing, in light of the previous recommendation precision results, we claim the following first findings:

– Regarding RQ1, *the **lda** aspect extraction method tends to improve the aspect-based recommender with which it is integrated*, although for cases rich in aspect opinions such as the Yelp datasets, the manually defined aspect vocabularies

Table 6 Coverage of the aspect extraction methods for every domain, using the abbreviations included in Table 4. Highest values for each method and dataset are in bold

Method	YELP						AMAZON									
	HOT		SPA		RES		MOV		MUS		CDS		PHO		GAM	
	K	%D	K	%D	K	%D	K	%D	K	%D	K	%D	K	%D	K	%D
voc	31	97.4	40	81.0	40	96.9	23	57.2	36	91.5	29	51.6	19	42.2	19	52.4
sab	2	82.4	1	20.4	0	0.0	1	53.8	2	85.2	2	73.1	2	60.6	1	83.0
sab	30	98.5	31	94.6	36	95.9	14	86.0	21	97.2	18	95.3	33	91.5	19	90.9
sab	64	99.2	63	97.3	85	98.1	30	91.7	50	98.3	41	97.3	64	93.9	47	94.7
sab	99	99.4	100	97.9	135	98.5	58	94.4	88	98.8	74	98.2	111	96.4	79	95.9
sab	278	**99.8**	273	**98.9**	339	**99.2**	223	**98.1**	278	**99.4**	262	**99.3**	264	**98.1**	229	**98.4**
dp	10	96.5	10	77.7	10	89.7	10	88.2	10	95.7	10	93.3	10	79.7	10	90.8
dp	50	98.8	50	91.9	50	96.8	50	96.8	50	98.4	50	97.5	50	91.7	50	95.4
dp	100	99.2	100	93.7	100	98.0	100	98.0	100	98.8	100	98.3	100	93.4	100	97.1
dp	200	99.4	200	94.8	200	98.6	200	98.6	200	99.1	200	98.8	200	94.6	200	98.0
dp	500	**99.5**	500	**96.0**	500	**99.0**	500	**99.0**	500	**99.3**	500	**99.1**	500	**95.4**	500	**98.7**
dpp	10	94.2	10	61.3	10	87.1	10	82.8	10	91.7	10	85.6	10	73.7	10	85.7
dpp	50	97.4	50	78.8	50	94.9	50	90.7	50	94.0	50	90.7	50	83.5	50	91.0
dpp	100	97.7	100	81.9	100	96.1	100	92.4	100	94.8	100	92.4	100	85.2	100	93.1
dpp	200	97.9	200	83.6	200	96.7	200	93.7	200	95.2	200	93.1	200	86.4	200	94.7
dpp	500	**98.1**	500	**84.6**	500	**97.2**	500	**94.5**	500	**95.7**	500	**93.7**	500	**87.3**	500	**95.7**

Table shows the number of aspects (K) and the percentage of reviews (%D) that have been annotated. Aspects for *SABRE* method correspond to $\varepsilon = \{0.1, 0.01, 0.005, 0.003, 0.001\}$ respectively. Coverage for *LDA* extraction method is 100% (and, hence, it is not reported)

obtained the best recommendation accuracy. The difference in behavior and performance of the evaluated recommenders could, to some extent, be attributed to the coverage of the annotations produced by each aspect extraction method (see Table 6). This might also explain why **lda**, which has full coverage in all the domains, represents in general a good aspect opinion extraction approach for recommendation purposes.
– With respect to RQ2, *the **cbub** hybrid aspect-based recommender effectively exploits aspect opinion information for all the tested domains*. Depending on the target domain, it either achieves the highest precision, or a precision very close to that of the best performing methods. At the end of this section, we shall compare the performance results of the aspect-based recommendation methods with those achieved by several baselines: standard recommenders that do not exploit aspect opinion information, and a state-of-the-art aspect-based recommender.

For a better understanding of the recommendation precision results, we analyze the behavior of the tested aspect extraction methods with respect to their parameters. In Fig. 2 we show the evolution of the precision achieved by all the methods on one domain of each dataset: Hotels (from Yelp) and Digital Music (from Amazon). We observed equivalent results with the remainder Yelp and Amazon datasets. As shown in the figure, the results are consistent with the previous analysis: **cbub** is the best performing method in both domains, and **cb** is the worst performing; the explicit user profiles obtain better results in HOT (and the other Yelp datasets), whereas the implicit user profiles perform better in MUS (and the other Amazon datasets). An interesting behavior that can be observed is how the sensitivity to the parameters changes in each method depending on the domain. On HOT there is a quite stable performance for the different parameter values, and the optimal parameters have small or medium values. In contrast, on MUS such stability is less clear, and the performance of the methods increases if we use smaller ε values for **sab**, larger N values for **dp** and **dpp**, and larger k values for **lda**. As commented previously, the aspect annotation coverage of the Yelp datasets were much lower than the Amazon datasets. The **lda** method, which obtains full coverage in all domains, performs quite stable varying its parameter values for both the Yelp and Amazon datasets.

To further address RQ2, we compare the accuracy of the proposed aspect-based recommendation methods with several baselines, presented in Sect. 5.4: non-personalized random (**rnd**) and item popularity-based (**ipop**) recommenders, standard content-based (**cb**) and collaborative filtering (**ib**, **ub** and **mf**) recommenders, and the state-of-the-art HFT (McAuley and Leskovec 2013) aspect-based recommender. In Table 7, we show the highest precision, recall and nDCG values achieved by all the methods. From the reported values (where all the differences are statistically significant using a Wilcoxon paired test with $p < 0.05$), we derive the following conclusions:

– The patterns of results and conclusions are equivalent for the three accuracy metrics.
– The **cbub** method outperforms all the baselines on the Yelp HOT and SPA domains, and is competitive with matrix factorization **mf** on the Yelp RES domain. We note that in these datasets, the average coverage of the manually defined aspects were around 90% of the available user reviews.

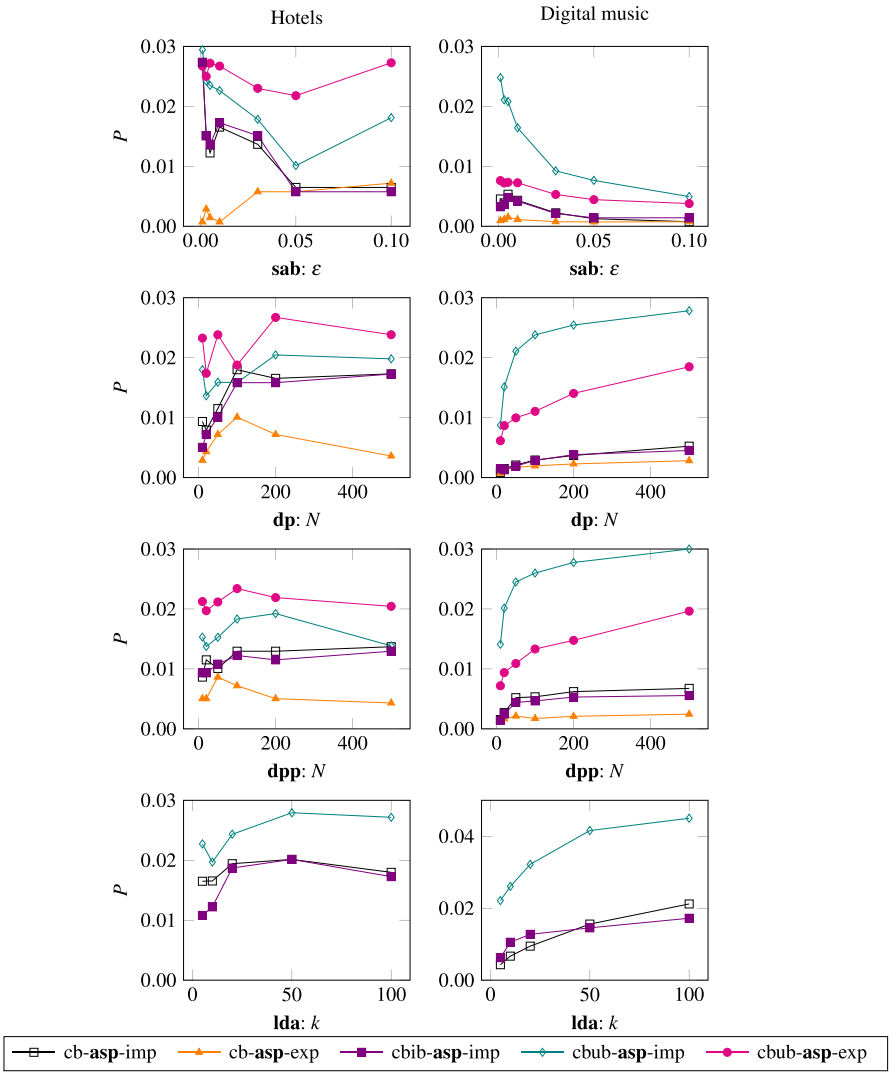

Fig. 2 Sensitivity of the recommendation performance in terms of P@5 for different parameters of the aspect extraction methods (**asp**), from top to bottom: sab, dp, dpp, and lda

– On the Amazon datasets, **cbub** outperforms the non-personalized and content-based methods, which shows that exploiting aspect opinion information is valuable even in cases where the aspect annotations have an average coverage of around 50% of the available user reviews.
– The user-based collaborative filtering **ub** methods is the most accurate on the Amazon datasets, whose users have relatively large numbers of ratings, as can be seen in Table 4.

Table 7 Comparison (baselines and best combinations of aspect-based recommenders) of performance values using precision, recall, and nDCG for every dataset

Metric	rec	YELP			AMAZON				
		HOT	SPA	RES	MOV	MUS	CDS	PHO	GAM
P	rnd	0.004	0.001	0.000	0.000	0.001	0.000	0.000	0.000
	ipop	**0.032**	**0.018**	0.009	0.003	0.007	0.002	0.005	0.004
	ib	0.011	0.009	0.013	0.023	0.050	0.025	0.012	0.021
	ub	0.025	0.015	0.012	**0.025**†	**0.055**†	**0.032**†	**0.018**†	**0.027**†
	mf	0.007	0.006	**0.013**†	0.015	0.052	0.018	0.016	0.024
	hft	0.006	0.002	0.000	0.000	0.001	0.000	0.000	0.000
	cb	0.027	0.009	0.007	0.006	0.021	0.009	0.002	0.005
	cbib	0.029	0.009	0.007	**0.015**	0.042	0.017	0.005	0.010
	cbub	**0.035**†	**0.023**†	**0.013**	0.015	**0.046**	**0.020**	**0.012**	**0.018**
R	rnd	0.020	0.006	0.001	0.000	0.001	0.000	0.000	0.000
	ipop	**0.156**	**0.086**	0.023	0.008	0.017	0.003	0.019	0.014
	ib	0.055	0.041	0.025	0.063	0.134	0.060	0.041	0.062
	ub	0.121	0.068	0.015	**0.064**†	**0.141**†	**0.074**†	**0.069**†	**0.081**†
	mf	0.032	0.028	**0.032**†	0.036	0.130	0.036	0.059	0.071
	hft	0.029	0.006	0.001	0.000	0.002	0.000	0.000	0.001
	cb	0.129	0.042	0.015	0.018	0.065	0.025	0.009	0.018
	cbib	0.135	0.040	0.014	0.039	0.106	0.036	0.016	0.028
	cbub	**0.171**†	**0.113**†	**0.030**	**0.039**	**0.123**	**0.046**	**0.042**	**0.052**
nDCG	rnd	0.010	0.003	0.001	0.000	0.001	0.000	0.000	0.000
	ipop	**0.095**	**0.050**	0.016	0.006	0.012	0.002	0.012	0.011
	ib	0.032	0.026	0.020	0.052	0.108	0.049	0.030	0.047
	ub	0.073	0.049	0.016	**0.054**†	**0.113**†	**0.062**†	**0.050**†	**0.062**†
	mf	0.019	0.019	**0.025**†	0.029	0.107	0.031	0.043	0.055
	hft	0.017	0.004	0.001	0.000	0.001	0.000	0.000	0.000
	cb	0.080	0.030	0.012	0.014	0.049	0.019	0.006	0.012
	cbib	0.078	0.027	0.011	**0.032**	0.084	0.030	0.011	0.021
	cbub	**0.108**†	**0.074**†	**0.022**	0.030	**0.099**	**0.038**	**0.031**	**0.039**

Best method for each domain and type of recommender algorithm in bold, best in domain for each metric marked with †

- The **hft** aspect-based baseline performs poorly. We note that this method is aimed to optimize the AUC metric, with which was evaluated in McAuley and Leskovec (2013), and thus is not expected to perform optimally on the item ranking task.
- On the Yelp HOT and SPA datasets, there is a bias on the items popularity, as can be seen by the high accuracy of **ipop** and the low accuracy of **mf**.

According to these observations, we can provide more details on the answer to RQ2, as well as first insights for RQ3, intended to understand how the type and coverage of extracted aspects affect the performance of aspect-based recommenders:

– Exploiting aspect opinion and rating data in a hybrid fashion as done by the **cbub** method allows achieving highly accurate recommendations in comparison to methods that only rely on content-based or rating data.
– In cases of high aspect annotation coverage ($\sim$ 90%), as in the Yelp datasets, **cbub** was the best (or almost the best) performing method, whereas in cases of low aspect annotation coverage ($\sim$ 50%), as in the Amazon datasets, the method outperformed non-personalized and CB baselines, and performed worse than **ub**, which was able to effectively exploit the relatively large amount of ratings per user.

6.2 Analyzing recommendation quality: coverage, novelty and diversity evaluation

In the recommender systems literature, it is well known that high accuracy in ranking metrics is difficult to balance with other evaluation dimensions, such as diversity and novelty (Zhou et al. 2010b). One paradigmatic example of this behavior is a recommender that suggests the most popular items: it usually shows high effectiveness at the expense of producing recommendations without diversity (the same popular items are *always* recommended) and novelty (usually considered as the inverse function of popularity).

Motivated by this issue, we aim to address RQ1 and RQ2 going beyond accuracy metrics. Hence, for the proposed aspect-based **cb**, **cbib** and **cbub** recommendation methods, we empirically compare their trade-offs between recommendation accuracy and several heterogeneous recommendation quality metrics. More specifically, we graphically report the USC (user coverage), AD (diversity) and EPC (novelty) values achieved by the evaluated methods, in comparison with their precision values. We show the graphic visualizations of user coverage in Fig. 3 and diversity and novelty in Figs. 4 and 5. We analyze next the tradeoffs for all these metrics.

Figure 3 shows USC values against $P@5$ values of the aspect-based recommenders on all the domains. We summarize the results as follows:

– In the Yelp datasets, the methods achieve the highest precision values. However, these values are obtained for a small (medium) percentage of the users on the HOT and SPA (and RES) domains.
– In the Amazon datasets, the methods do not achieve the highest precision values in comparison with collaborative filtering. However, these values are obtained for all the available users on all the domains.
– For every domain, the **cbub** method achieves the best tradeoff between precision and user coverage: it tends to be located further on the right (higher $P@5$) and top (higher USC) in the visual representations of each figure row, i.e., of each domain.
– In terms of precision-coverage trade-off, there is no clear winner among the aspect extraction methods, but for **cbub** with implicit user profiling, **lda** is the best performing one, followed by **dpp** and **sab**.

Figures 4 and 5 respectively show AD and EPC values against $P@5$. In these cases, we observe the same result patterns for both diversity and novelty metrics, so we jointly summarize the conclusions derived from them as follows:

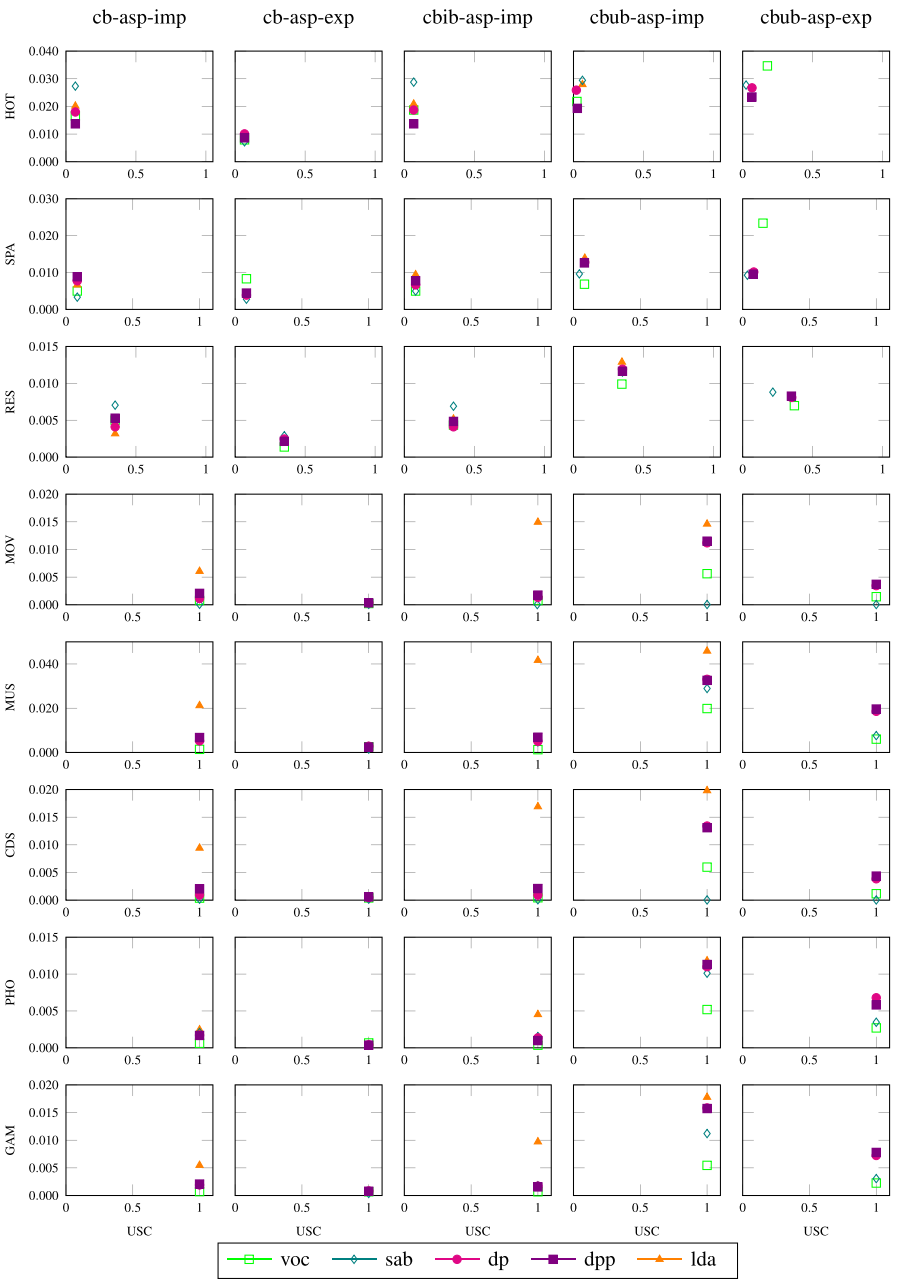

Fig. 3 Trade-offs between recommendation coverage (USC) and precision ($P@5$) for different recommendation and user profile strategies, from left to right: cb-asp-imp, cb-asp-exp, cbib-asp-imp, cbub-asp-imp, and cbub-asp-exp, where asp is one of the 5 aspect extraction methods: voc, sab, dp, dpp, and lda

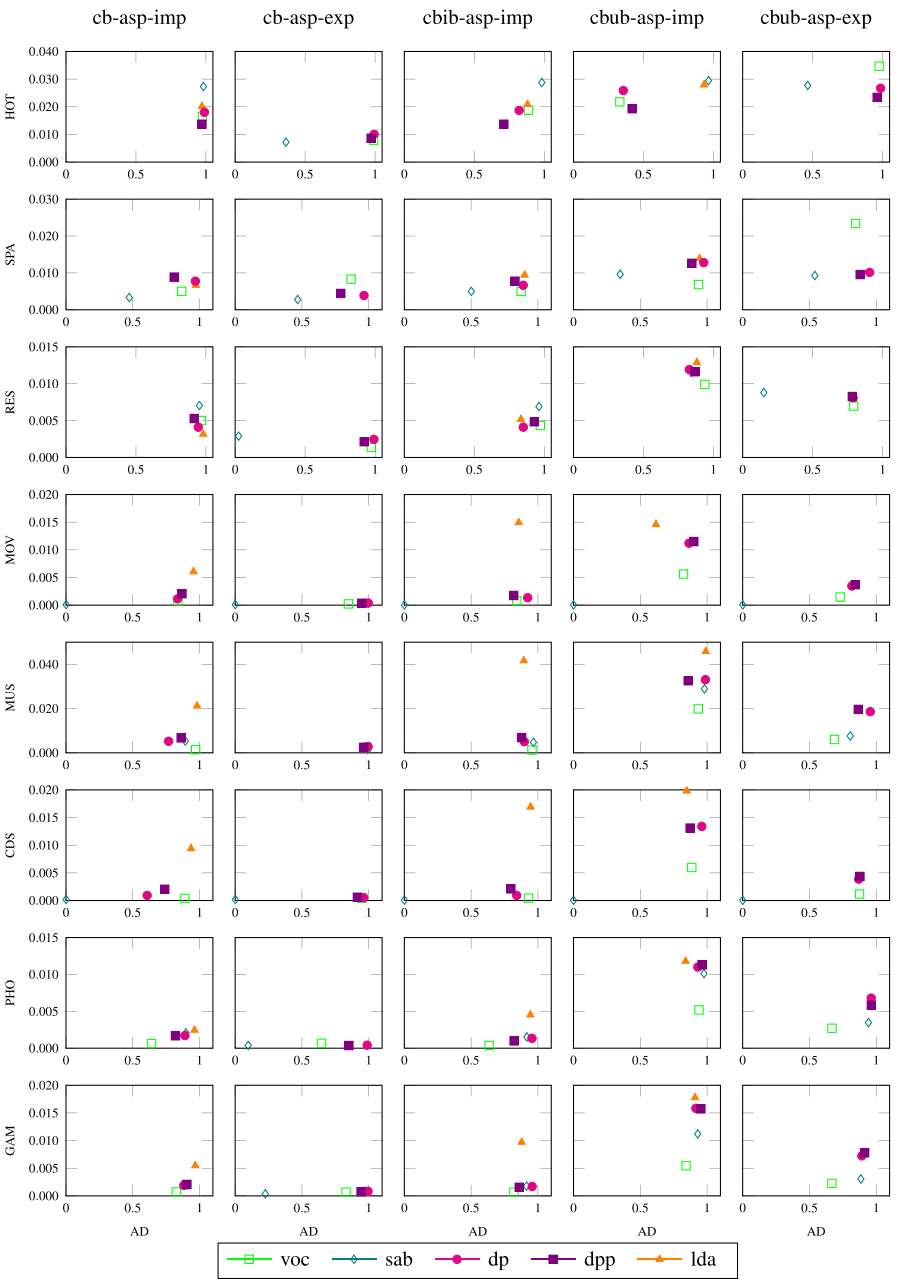

Fig. 4 Trade-offs between recommendation diversity (AD) and precision ($P@5$) for different recommendation and user profile strategies, from left to right: cb-asp-imp, cb-asp-exp, cbib-asp-imp, cbub-asp-imp, and cbub-asp-exp, where asp is one of the 5 aspect extraction methods: voc, sab, dp, dpp, and lda

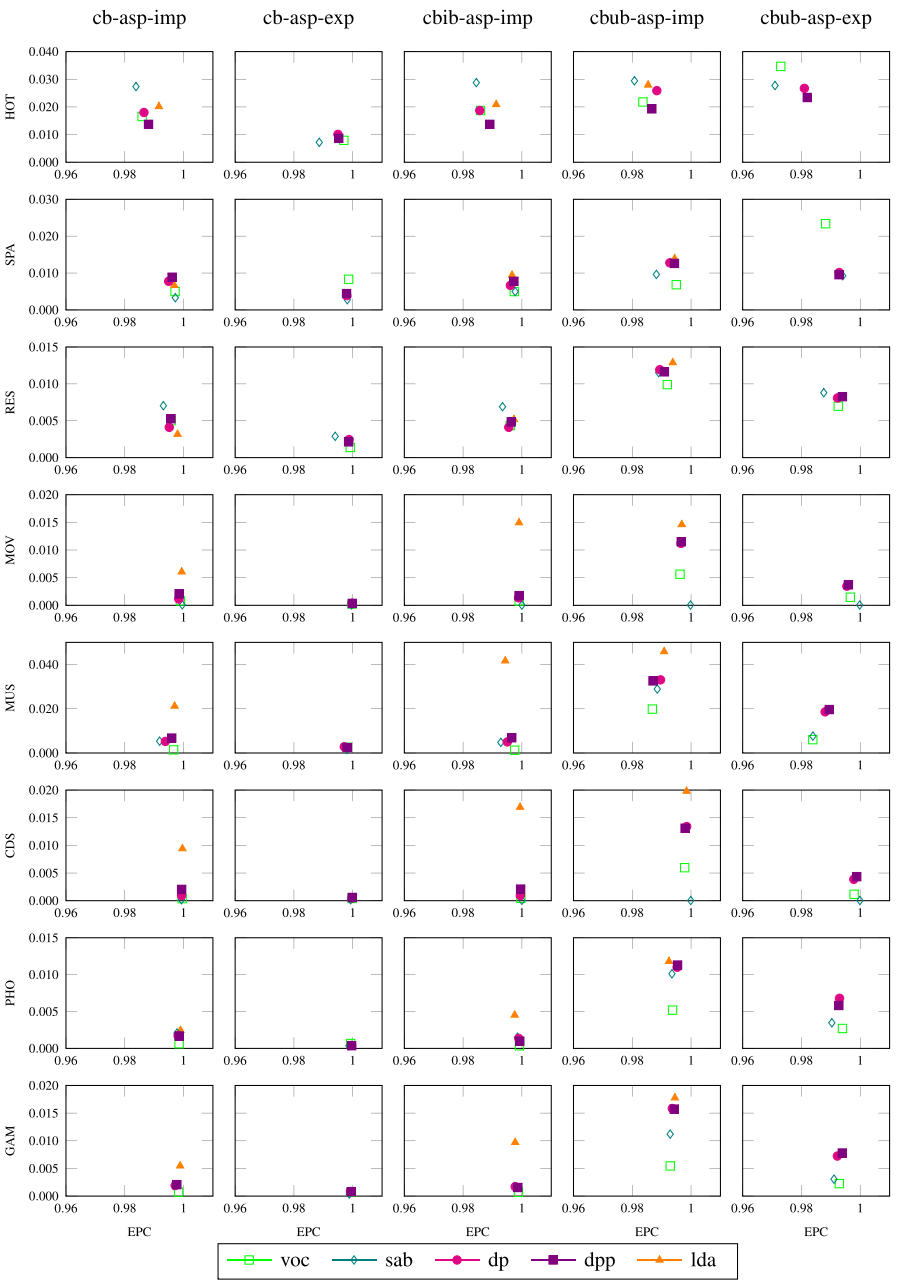

Fig. 5 Trade-offs between recommendation novelty (EPC) and precision ($P@5$) for different recommendation and user profile strategies, from left to right: cb-asp-imp, cb-asp-exp, cbib-asp-imp, cbub-asp-imp, and cbub-asp-exp, where asp is one of the 5 aspect extraction methods: voc, sab, dp, dpp, and lda

– The **cbub** aspect-based recommender with implicit user profiles achieves the best trade-offs between precision and diversity/novelty on all the domains when **lda** aspect annotations are exploited. This also occurs when the recommender exploits other aspect annotations, except on the HOT domain, where the aspects extracted by **dp** and **dpp** obtained less diverse recommendations.
– In general, the recommendations generated on the Yelp domains were more diverse and novel than on the Amazon datasets, for all aspect extraction and aspect-based recommendation methods.

These conclusions can be considered as additional arguments for our answers to RQ1 and RQ2, which state that (i) the **lda** method generates data consistently effective for both content-based and collaborative filtering, (ii) aspect-based recommendations generated by **cbub** are of high quality in terms of both accuracy metrics and trade-offs between accuracy and non-accuracy metrics, such as user coverage, ranking diversity and item novelty, and (iii) the percentage of reviewed items annotated with aspect opinions (e.g., $\sim$ 90% in Yelp datasets and $\sim$ 50% in Amazon) is critical to improve personalized recommendations generated with only rating data. In the next section, we further analyze the effect of the coverage and type of extracted aspects on the performance of aspect-based recommendation methods (RQ3).

6.3 Analyzing the recommendation effects of aspect types and annotation coverage

Our research question RQ3 focuses on understanding the effects that the type and coverage of aspect opinions may have on the performance of recommendation methods that exploit them. In Sects. 6.1 and 6.2, we provided first insights about the importance of having a high coverage of annotated items in order to build good performing aspect-based recommenders. In this section, we present a number of additional experiments and analysis aimed to further address such question.

So far, we have not made any assumption about the collected and exploited aspect-based data. We have used all the available ratings, and all the aspect opinion annotations provided by the extraction methods for the user reviews of the datasets. However, in the datasets, not all the reviews actually have personal opinions on item aspects. Moreover, the annotation processes are not perfect, and are not able to capture all the existing aspect opinions due to noun coreferences, misspellings, slang language and word abbreviations, among other issues. This entails that many items may have assigned none or a few aspect opinion annotations.

In this section, we analyze the potential problem of such a situation, by simulating an appropriate scenario for aspect-based recommendation methods: we shall run the experiments only on those reviews with at least one aspect opinion annotation (from the most restrictive *voc* method). Although limited, this would approximate the ideal situation where any aspect opinion extraction method has full annotation coverage. Additionally, we also analyze how existing aspect types—namely explicit and implicit—have some non-performance implications in recommendation tasks.

6.3.1 Analyzing the coverage of the aspect extraction methods

In Table 6, we showed the initial coverage achieved by each extraction method in the conducted experiments. This measure accounts for the ability of a method to find at least one aspect opinion in each review. In the table we included the number of aspects annotated (K) and the percentage of reviews with at least one annotation (%D). We note that the **lda** method was not included in the table, since it is able to generate latent topics (not necessarily aspects) for every item on all domains.

We observe that, as expected, the **voc** vocabulary-based method—which uses an initial, manually defined list of seed terms as aspects—achieves a significantly smaller coverage than the other methods, with a similar number of aspects (except on the Restaurants domain). For example, on the Cell Phones domain, it obtains a coverage of 42.2% with 19 aspects, whereas a value above 90% is achieved by SABRE (**sab**) and Double Propagation (**dp** and **dpp**).

In this context, it has to be noted that, differently to **voc**, the **sab** and **dp** methods are able to capture more terms as aspects, but generate some annotations that do not refer to actual aspects (see Table 10 in the "Appendix" of the paper). Regarding other aspect extraction methods, we observe that the coverage of **dp** is higher than that of **dpp**. This is an expected result, since frequent but not useful nouns are removed in the pruning stage of **dpp**.

6.3.2 Analyzing situations with full coverage of aspect annotation

Based on the coverage results presented in the previous subsection, we would expect that some recommendation performance changes may occur when the simulated *ideal* scenario described above is compared against the original one, where all the reviews are considered and there are items without aspect opinion annotations. More specifically, we would expect to obtain larger accuracy improvements by the aspect-based recommendation methods, to the detriment of rating-based collaborative filtering methods. However, as we shall see next, this is not always the case.

Table 8 shows the performance (in terms of P@5) achieved when considering only the (user, item) pairs whose items have at least one aspect opinion. For the sake of simplicity, we only report the values for two domains (Digital Music and Hotels). We also include the performance improvement with respect to the original scenario, that is, the (positive or negative) improvement with respect to the values reported in Table 5.

In the Amazon MUS domain we observe that, as expected, the collaborative filtering methods decrease their performance. However, not all the aspect-based recommendation methods show significant improvements. The largest (global) improvement is obtained for **cb**, the pure content-based recommendation method. This makes sense, since the simulated scenario is aimed at favoring this type of algorithms. In fact, this behavior is also observed in the HOT domain.

In the Yelp HOT domain, the conclusions are less clear: only few aspect-based recommenders improve their performance. Moreover, some of the collaborative filtering methods evidence a performance increase. This is something that might be attributed

Table 8 Comparison of performance values (measured as P@5) for the Digital music (mus) and Hotel (hot) domains

rec	asp	up	HOT	ΔHOT	MUS	ΔMUS
ib	–	–	0.014	**26.2**	0.034	− 32.5
ub	–	–	**0.021**	− 13.3	**0.044**†	− 19.0
mf	–	–	0.008	19.7	0.044	**− 16.5**
cb	voc	imp	0.014	− 13.7	0.001	− 6.2
cb	voc	exp	0.013	**67.6**†	0.003	3.4
cb	sab	imp	**0.017**	− 36.5	0.005	− 9.7
cb	sab	exp	0.011	55.7	0.001	− 9.9
cb	dp	imp	0.017	− 3.6	0.005	− 5.2
cb	dp	exp	0.005	− 49.3	0.003	6.6
cb	dpp	imp	0.011	− 18.1	0.005	− 24.5
cb	dpp	exp	0.007	− 17.3	0.003	**11.2**
cb	lda	imp	0.016	− 19.0	**0.019**	− 11.7
cbib	voc	–	0.013	− 28.9	0.002	**29.7**†
cbib	sab	–	0.016	− 43.3	0.004	− 9.0
cbib	dp	–	**0.018**	**− 1.8**	0.005	− 5.6
cbib	dpp	–	0.009	− 33.0	0.005	− 26.0
cbib	lda	–	0.016	− 21.7	**0.037**	− 10.3
cbub	voc	imp	0.015	− 31.5	0.015	− 23.4
cbub	voc	exp	0.030	− 13.8	0.005	− 12.1
cbub	sab	imp	0.025	− 15.1	0.021	− 26.9
cbub	sab	exp	0.024	− 14.5	0.008	**3.2**
cbub	dp	imp	0.016	− 38.2	0.024	− 26.5
cbub	dp	exp	0.027	− 0.7	0.015	− 17.7
cbub	dpp	imp	0.014	− 30.0	0.027	− 18.2
cbub	dpp	exp	0.019	− 20.7	0.018	− 6.2
cbub	lda	imp	**0.030**†	**7.2**	**0.037**	− 19.6

The column denoted with Δ shows the performance improvement with respect to the unfiltered data (Table 5), that is, $\Delta = (m_2 - m_1)/m_1$ where m_2 is the new measurement and m_1 the previous one. Highest values in each column are denoted with a †

to a larger coverage for this domain originally (as shown in Table 6), which, in turn, creates a constrained dataset very similar to the original one.

Based on these observations, we provide a more detailed answer to RQ3. A higher coverage of items annotated with aspect opinions may have a positive effect on recommendation performance, as shown for the **cb** method. Improvements on such performance, on the other hand, may also depend significantly on the amount of available ratings for those recommendation methods that exploit both aspect opinions and ratings, as done by the **cbub** method.

6.3.3 Analyzing the types of extracted aspects

In the "Appendix" of the paper, Table 10 shows a qualitative comparison of the most frequent explicit aspects extracted from the user reviews on each domain. For the Double Propagation methods, **dp** and **dpp**, we show the top $N = 20$ aspects in decreasing order of frequency. For the SABRE method (**sab**) we show those aspects with a score higher than $\varepsilon = 0.01$, in decreasing order of score as well.

In general, we observe that the different methods do have many meaningful aspects in common, meaning that all of them are suitable for the aspect opinion extraction task, even when each method works from a particular perspective. We also note that both **sab** and **dp** consider as aspects some noisy terms, such as *one* or *anyone* in HOT.

The effect of the pruning stages for Double Propagation can be characterized in two different ways. On the one hand, **dpp** removes common nouns that appear in sentences together with other nouns. For example, in the Video Games domain, it removes *a* and *year* from the top list. On the other hand, it is able to identify compound noun aspects, such as *screen protector*.

In general, these methods extract meaningful aspects. We observe that some wrong annotations refer to e.g. proper nouns, prepositions, determinants and adverbs, which could be easily filtered out. Other annotations, in contrast, are nouns related to domain topics that do not correspond to aspects. Dealing with these annotation cases for recommendation purposes is something worth investigating.

Besides the explicit aspects extracted by the above methods, the implicit aspects generated by the **lda** method have to be considered as well. In our experiments, we have shown that this type of aspect annotations allows for better performance of the recommenders. However, these annotations, which do not necessarily correspond to real aspects, but to domain topics or other concepts, do not allow providing the user with explanations about the generated recommendations. This also represents a difficulty for making multi-criteria or constrained recommendations, which are based on references to explicit, legible aspects.

Summarizing, and further addressing RQ3, we conclude that (i) there is a significant overlap between the sets of explicit aspects extracted by **sab** and **dp**, (ii) some wrong aspects extracted by these methods could be easily handled considering simple grammatical and syntactical issues, and (iii) the implicit aspects extracted by **lda** obtain the best performing recommendations, but limit the explainability of such recommendations.

7 Conclusions and future work

In this paper, we have presented an exhaustive evaluation combining a number of methods to extract aspect opinions from reviews, and methods that exploit such information to provide personalized item recommendations. Both the aspect opinion extraction and aspect-based recommendation methods are representative examples of the different approaches existing in the research literature. This, together with the facts that we have analyzed heterogeneous metrics (i.e., precision, recall, nDCG, coverage, diversity and novelty) on large datasets from Yelp and Amazon systems for several domains (hotels,

restaurants, movies, music and mobile phones, among others), and have considered different characteristics of the datasets, domains and extracted aspects (e.g., nature and purpose of the source systems, amounts of ratings per user, and coverage of items annotated with aspect opinions), have allowed us to give argued conclusions about the stated research questions.

In particular, according to our experimental results, we have shown that the aspects extracted by **lda**, a topic model-based method that represents items in terms of the topics discussed in their reviews, resulted the most effective for recommendation purposes in general (RQ1). We have also seen that **cbub**—a proposed aspect-based hybrid recommender that incorporates aspect opinion information into the user-based collaborative filtering heuristic—consistently generates effective recommendations, outperforming standard baselines (RQ2). Depending on the target domain, the combination of **lda** and **cbub** either achieves the highest precision, or a precision very close to that of the best recommender, and maintains a good tradeoff between recommendation accuracy and recommendation diversity and novelty. Moreover, in general, we have observed that the coverage of the aspect opinion extraction methods has an important impact on the recommendation performance (RQ3). We have identified differences between domains with high coverage ($\sim$ 90% in Yelp datasets) and domains with low coverage ($\sim$ 50% in Amazon datasets).

In this context, although **lda** has outperformed the other methods thanks to its better generalization capabilities, it has to be noted that not all the topics generated by this method correspond to item aspects, and that such topics represent implicit (latent) semantic concepts. This limits the interpretation of the extracted aspects by end users, and their applicability to explain generated recommendations (Chen and Wang 2014). For this reason, we believe that further research should be done in this line.

As recently done by Musto et al. (2014), we have explored simple, yet effective hybrid recommendation methods that, within the collaborative filtering heuristic framework, exploit effectively user preferences for item aspects. In our experiments, consistently on the domains of Yelp datasets, these methods have outperformed state-of-the-art CF approaches, including those based on Matrix Factorization. Nonetheless, as future work, we propose to investigate MF models designed to exploit aspect opinion information. There is a great amount of work aimed to incorporate side information into matrix factorization for recommendation; see e.g. Gunawardana and Meek (2009), Pilászy at al. (2015), Chen et al. (2011). Regarding information about aspect opinions, we find interesting to investigate approaches like LRPPM, the learning-to-rank tensor-matrix factorization framework proposed by Chen et al. (2016). This framework aims to learn user preferences for features at both item and item category levels, by modeling interactions between users, items and aspects simultaneously. Evaluated on subsets of the Yelp and Amazon datasets used in this paper, LRPPM achieved nDCG values comparable to those reported in this paper.

In addition to Matrix Factorization, we also believe that Deep Learning represents a promising approach to aspect-based recommendation. Deep learning uses a cascade of multiple layers of nonlinear processing units for feature extraction and transformation, and are able to learn multiple levels of representations. The modularity of neural network architectures also allows handling heterogeneous, unstructured data, such as text content. In the context of recommender systems (Rendle et al. 2009; Van den

Oord et al. 2013; He and McAuley 2016), deep learning is gaining momentum due to its state-of-the-art performance (Zhang et al. 2017), and its capability to provide a better understanding of user preferences, item characteristics, and interactions between them. In particular, we envision the combination of deep learning and word embedding techniques as an effective approach to extract aspect opinions from text, and further exploit them for recommendation purposes.

Regardless the followed algorithmic approach to aspect-based recommendation, as stated by Chen et al. (2015), in addition to global ratings and aspect opinions, other elements of user reviews could be exploited to enhance item recommendations, such as the reviewers' expertise and the aspects popularity. Among these elements, contextual conditions represent a very valuable source of information. For instance, in hotel recommendations, the aspects *cleanliness* and *price* may be the most important aspects for a user who is planning one-week holidays (*period of time* context) with his/her family (*companion* context). Levi et al. (2012) motivated this issue through a user study on the hotel recommendation domain, by considering user intent and background (nationality) as contextual dimensions. More recently, Chen and Chen (2014) proposed a recommendation method exploiting co-occurrences of aspect opinions and context values in reviews. The authors, however, performed a very simple keyword matching technique to extract context information, and did not report which context dimensions they used in their experiments. Existing work is thus not mature yet and, in our humble opinion, this is an interesting and relevant research topic.

On certain domains, such as hotels and restaurants, there are issues about time and location which result challenging for future work. A user's preferences can change over time, so the time frame of the reviews should be considered. Similarly, the user's current location has to be carefully considered with respect to previous locations and their associated user preferences. Hence, context-aware user modeling for aspect-based recommendation is an open research line.

Finally, it has to be noted that in our experiments, we did not analyze the amount of aspect opinion data required to achieve a certain level of performance on the recommendation tasks. The user preference scarcity, commonly referred to as cold-start, and sparsity are well-known CF problems, which also apply to the aspect-based recommendation methods (Levi et al. 2012; Chen and Wang 2013), a fact that, to the best of our knowledge, has not been investigated in depth yet. In this sense, we also understand that a deeper analysis of the results with respect to other characteristics of the datasets could be done to better show the generalizability of our findings. We leave this as future work.

Acknowledgements This work was supported by the Spanish Ministry of Economy, Industry and Competitiveness (TIN2016-80630-P). The authors thank the reviewers for their thoughtful comments and suggestions.

A Appendix

For the sake of reproducibility, in Table 9 we present the optimal parameter values found for the recommendation methods presented in Sect. 6, and, specifically, for the results reported in Tables 5 and 7.

These parameters were obtained by running all the possible method combinations, and selecting the best performing ones according to P@5. In particular, a grid search was conducted based on the following values of the parameters:

- Number of neighbors (rec column for ub, cbib, and cbub): 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100.
- Number of latent factors (rec column for mf): 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100.
- Threshold to select terms (met column when asp is sab): 0.1, 0.05, 0.03, 0.01, 0.005, 0.003, 0.001.
- Top terms (met column when asp is dp or dpp): 10, 20, 50, 100, 200, 500.
- Number of latent topics (met column when asp is lda): 5, 10, 20, 50, 100.
- Maximum number of words from the corpus (rec column for hft): 5 K, 50 K, 500 K. The regularizers for the latent topic (0, 0.1, 0.5) and MF (0.1, 0.5, 1) as well as the number of latent factors/topics (5, 10) were also tested but no important differences were observed, as in the original paper; hence, 0, 0.1 and 5 were used for these parameters in every dataset.

Note that the non-personalized techniques such as **rnd** and **ipop** do not use any parameter (denoted as − in the table); furthermore, pure collaborative filtering algorithms (**ib**, **ub**, **mf**) do not need any parameter regarding the aspect extraction method because they do not exploit aspect opinion information. It should also be noted that the **cb** pure content-based method and the **voc** vocabulary-based aspect extraction method do not have parameters either. Additionally, as a representative example, in Table 10 we show the extracted aspects by the Double Propagation and SABRE methods using top 20 terms and 0.01 threshold, respectively.

Table 9 Parameter values of the recommenders (rec column) and aspect extraction methods (asp column) whose results are reported in Tables 5 and 7

rec	asp	up	YELP						AMAZON									
			HOT		SPA		RES		MOV		MUS		CDS		PHO		GAM	
			rec	met	rec	met	rec	met	rec	met	rec	met	rec	met	rec	met	rec	met
rnd	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
ipop	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
ib	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
ub	–	–	100	–	50	–	5	–	100	–	100	–	100	–	100	–	100	–
mf	–	–	5	–	10	–	15	–	50	–	50	–	50	–	50	–	50	–
hft	–	–	50 K	–	50 K	–	50 K	–	500 K	–	50 K	–	5K	–	5K	–	5K	–
cb	voc	imp	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
cb	voc	exp	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
cb	sab	imp	–	0.001	–	0.05	–	0.003	–	0.01	–	0.005	–	0.03	–	0.003	–	0.005
cb	sab	exp	–	0.1	–	0.05	–	0.1	–	0.001	–	0.005	–	0.001	–	0.1	–	0.1
cb	dp	imp	–	100	–	500	–	500	–	500	–	500	–	500	–	500	–	500
cb	dp	exp	–	100	–	50	–	500	–	500	–	500	–	500	–	500	–	500
cb	dpp	imp	–	500	–	200	–	500	–	500	–	500	–	500	–	500	–	500
cb	dpp	exp	–	50	–	100	–	100	–	500	–	500	–	500	–	500	–	500
cb	lda	imp	–	50	–	50	–	50	–	100	–	100	–	100	–	100	–	100
cbib	voc	–	10	–	50	–	100	–	100	–	15	–	100	–	100	–	50	–
cbib	sab	–	50	0.001	100	0.05	100	0.003	100	0.001	100	0.005	100	0.03	100	0.003	50	0.005
cbib	dp	–	10	500	10	100	10	500	15	500	10	500	50	500	50	500	100	500
cbib	dpp	–	5	100	50	200	50	500	15	500	15	500	15	500	100	200	15	500
cbib	lda	–	10	50	15	100	5	100	5	100	5	100	10	100	10	100	5	100
cbub	voc	imp	5	–	100	–	50	–	100	–	50	–	100	–	100	–	100	–
cbub	voc	exp	100	–	100	–	100	–	100	–	100	–	100	–	100	–	100	–

Table 9 continued

rec	asp	up	YELP						AMAZON									
			HOT		SPA		RES		MOV		MUS		CDS		PHO		GAM	
			rec	met	rec	met	rec	met	rec	met	rec	met	rec	met	rec	met	rec	met
cbub	sab	imp	100	0.001	50	0.1	100	0.001	5	0.001	15	0.001	15	0.05	50	0.001	50	0.001
cbub	sab	exp	50	0.05	50	0.03	100	0.05	5	0.001	100	0.001	15	0.003	100	0.001	100	0.001
cbub	dp	imp	5	200	100	100	100	500	50	500	15	500	15	500	100	500	50	500
cbub	dp	exp	100	200	100	200	100	500	100	500	50	500	100	500	100	500	100	500
cbub	dpp	imp	5	20	100	100	100	500	50	500	50	500	50	500	100	500	50	500
cbub	dpp	exp	100	100	100	200	100	500	100	500	100	500	100	500	100	200	100	500
cbub	lda	imp	100	50	100	100	50	50	100	100	15	100	50	100	100	100	50	100

They correspond to the optimal values obtained for each combination of recommendation (rec), item profiling (asp), and user profiling (up) methods, with respect to the P@5 metric

Table 10 Extracted aspects with *Double Propagation* and *SABRE*

Domain	Method	Aspects
HOT	sab	Room, hotel, pool, resort, phoenix, place, staff, breakfast, stay, spa, restaurant, night, desk, bed, bar, lobby, something, service, nothing, everyone, bathroom, hilton, location, parking, internet, free, area, shower, food, ho
	dp	Room, hotel, stay, pool, place, staff, service, time, night, area, bed, breakfast, day, bar, one, restaurant, food, resort, desk, thing
	dpp	Room, hotel, pool, place, staff, service, night, time, area, bed, bar, day, resort, breakfast, food, restaurant, stay, desk, people, front desk
SPA	sab	Massage, pedicure, spa, nail, salon, hair, place, color, manicure, something, haircut, room, everyone, stylist, service, pool, anyone, barber, time, resort, appointment, polish, experience, staff, cut, nothing, today, location, price, phoenix, store
	dp	Place, time, service, massage, staff, experience, price, job, room, salon, spa, day, nail, pedicure, hair, pool, year, one, area, thing
	dpp	Place, time, massage, service, room, spa, nail, salon, staff, pool, hair, pedicure, day, experience, resort, area, job, hotel, haircut, store
RES	sab	Food, place, pizza, menu, flavor, restaurant, something, chicken, burger, salad, sauce, sushi, taco, cheese, sandwich, nothing, lunch, appetizer, phoenix, service, salsa, dish, everyone, drink, meal, bar, server, burrito, beer, dinner, dessert, waitress, rice, table, patio, meat
	dp	Food, place, service, time, order, restaurant, one, menu, price, a, great, chicken, try, love, thing, drink, salad, not, table, sauce
	dpp	Food, place, time, service, restaurant, menu, chicken, salad, lunch, bar, sauce, cheese, table, meal, night, thing, drink, order, people, pizza
MOV	sab	Movie, film, something, story, character, scene, anyone, episode, everyone, nothing, show, actor, john, plot
	dp	Movie, film, one, time, the, this, a, story, character, love, not, dvd, way, show, scene, end, watch, thing, other, year
	dpp	Movie, film, time, story, character, way, people, show, scene, series, life, love, action, season, plot, dvd, man, episode, thing, family
MUS	sab	Album, song, cd, track, music, lyric, something, band, vocal, beat, fan, hit, guitar, rock, nothing, love, rap, sound, anyone, pop, ballad
	dp	Album, song, track, music, time, one, sound, cd, love, lyric, this, fan, year, way, band, the, a, release, rock, work
	dpp	Album, song, music, track, time, band, cd, sound, love, rock, way, guitar, beat, voice, rap, title track, record, hit, work, one
CDS	sab	Album, song, cd, music, track, band, something, lyric, fan, vocal, guitar, rock, nothing, anyone, sound, recording, favorite, hit
	dp	Album, song, music, cd, sound, time, one, track, band, fan, love, this, a, the, year, rock, way, work, release, voice
	dpp	Album, song, music, band, cd, time, track, sound, rock, guitar, love, voice, way, work, fan, metal, version, record, one, people
PHO	sab	Phone, case, charger, battery, screen, protector, device, headset, color, cable, product, button, something, galaxy, %, app, port, headphone, amazon, anyone, quality, cover, price, stylus, adapter, cord, fit, protection, review, nexus, rubber, ear, charge

Table 10 continued

Domain	Method	Aspects
	dp	Phone, case, use, one, product, time, charge, screen, fit, iphone, work, price, battery, look, charger, quality, device, thing, protector, not
	dpp	Phone, case, screen, product, battery, time, charger, price, device, screen protector, iphone, protector, quality, charge, color, cable, protection, button, use, car
GAM	sab	Game, graphic, fun, gameplay, something, character, gamer, story, multiplayer, nothing, controller, anyone, mode, enemy, player, everyone, gaming, mission, fan
	dp	Game, play, time, fun, one, graphic, thing, way, a, story, character, lot, other, this, gameplay, level, use, not, people, player
	dpp	Game, time, fun, story, character, thing, way, gameplay, level, play, people, lot, player, system, great game, mode, enemy, one, controller, weapon

For **dp** and **dpp**, we show top $N = 20$ most frequent aspects sorted in descending order, and for **sab**, those aspects with a score above $\varepsilon = 0.01$, also presented in descending score value

References

Aciar, S., Zhang, D., Simoff, S., Debenham, J.: Informed recommender: basing recommendations on consumer product reviews. IEEE Intell. Syst. **22**(3), 37–47 (2007)

Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.). Recommender Systems Handbook, pp. 191–226. Springer (2015)

Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)

Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB'94, pp. 487–499 (1994)

Bafna, K., Toshniwal, D.: Feature based summarization of customers reviews of online products. Procedia Comput. Sci. **22**, 142–151 (2013)

Bauman, K., Liu, B., Tuzhilin, A.: Aspect based recommendations: recommending items with the most valuable aspects based on user reviews. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'17, pp. 717–725. ACM (2017)

Bellogín, A., Castells, P., Cantador, I.: Precision-oriented evaluation of recommender systems: an algorithmic comparison. In: Proceedings of the 5th ACM Conference on Recommender Systems, RecSys'11, pp. 333–336. ACM (2011)

Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)

Caputo, A., Basile, P., de Gemmis, M., Lops, P., Semeraro, G., Rossiello, G.: SABRE: a sentiment aspect-based retrieval engine. In: Information Filtering and Retrieval, pp. 63–78. Springer (2017)

Castells, P., Hurley, N.J., Vargas, S.: Novelty and diversity in recommender systems. In: Recommender Systems Handbook, pp. 881–918. Springer (2015)

Chen, G., Chen, L.: Recommendation based on contextual opinions. In: Proceedings of the 22nd Conference on User Modelling, Adaptation and Personalization, UMAP'14, pp. 61–73. Springer (2014)

Chen, D., Manning, C.: A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP'14, pp. 740–750. ACL (2014)

Chen, L., Wang, F.: Preference-based clustering reviews for augmenting e-commerce recommendation. Knowl. Based Syst. **50**, 44–59 (2013)

Chen, L., Wang, F.: Sentiment-enhanced explanation of product recommendations. In: Proceedings of the 23rd International Conference on World Wide Web, WWW'2014, pp. 239–240. ACM (2014)

Chen, T., Zheng, Z., Lu, Q., Zhang, W., Yu, Y.: Feature-based matrix factorization (2011). arXiv preprint arXiv:1109.2271

Chen, L., Chen, G., Wang, F.: Recommender systems based on user reviews: the state of the art. User Model. User Adapt. Interact. **25**(2), 99–154 (2015)
Chen, X., Qin, Z., Zhang, Y., Xu, T.: Learning to rank features for recommendation over multiple categories. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'16, pp. 305–314. ACM (2016)
Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the 4th ACM Conference on Recommender Systems, RecSys'10, pp. 39–46. ACM (2010)
Diao, Q., Qiu, M., Wu, C.Y., Smola, A.J., Jiang, J., Wang, C.: Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14. ACM (2014)
Dong, R., Schaal, M., O'Mahony, M.P., McCarthy, K., Smyth, B.: Opinionated product recommendation. In: Proceedings of the 21st International Conference on Case-Based Reasoning, ICCBR'13, pp. 44–58. Springer (2013)
Esuli, A., Sebastiani, F.: SentiWordNet: a high-coverage lexical resource for opinion mining. Evaluation **17**, 1–26 (2007)
Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: an experimental study. Artif. Intell. **165**(1), 91–134 (2005)
Ganu, G., Kakodkar, Y., Marian, A.: Improving the quality of predictions using textual information in online user reviews. Inf. Syst. **38**(1), 1–15 (2013)
García Esparza, S., O'Mahony, M.P., Smyth, B.: A multi-criteria evaluation of a user generated content based recommender system. In: Proceedings of 3rd Workshop on Recommender Systems and the SocialWeb, RSWEB'11 (2011)
Gunawardana, A., Meek, C.: A unified approach to building hybrid recommender systems. In: Proceedings of the 4th ACM Conference on Recommender Systems, RecSys'09, pp. 117–124. ACM (2009)
He, R., McAuley, J.: VBPR: visual Bayesian personalized ranking from implicit feedback. In: Proceedings of the 13th AAAI Conference on Artificial Intelligence, AAAI'16, pp. 144–150 (2016)
Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: Analgorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'99, pp. 230–237. ACM (1999)
Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Mach. Learn. **42**(1), 177–196 (2001)
Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'04, pp. 168–177. ACM (2004a)
Hu, M., Liu, B.: Mining opinion features in customer reviews. In: Proceedings of the 19th National Conference on Artificial Intelligence, AAAI'14, pp. 755–760. AAAI Press (2004b)
Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 8th IEEE International Conference on Data Mining, ICDM'08, pp. 263–272. IEEE (2008)
Jakob, N., Gurevych, I.: Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP'10, pp. 1035–1045. ACL (2010)
Jakob, N., Weber, S.H., Müller, M.C., Gurevych, I.: Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. In: Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion, TSA'09, pp. 57–64. ACM (2009)
Jamroonsilp, S., Prompoon, N.: Analyzing software reviews for software quality-based ranking. In: Proceedings of the 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON'10, pp. 1–6. IEEE (2013)
Jannach, D., Adomavicius, G.: Recommendations with a purpose. In: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys'16, pp. 7–10. ACM (2016)
Ko, M., Kim, H.W., Mun, Y.Y., Song, J., Liu, Y.: Moviecommenter: aspect-based collaborative filtering by utilizing user comments. In: Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom'11, pp. 362–371. IEEE (2011)
Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)

Kumar, K.P.V., Kumar, G.S.C., Aruna, M., Srinivas, B.: Mining online customer reviews for product feature-based ranking. Int J Adv Res Comput Sci **6**(3), 23–27 (2015)
Kumar, S., Gao, X., Welch, I.: Co-clustering for dual topic models. In: Proceedings of the 9th Australasian Joint Conference on Advances in Artificial Intelligence, AI'16, pp. 390–402. Springer (2016)
Levi, A., Mokryn, O., Diot, C., Taft, N.: Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In: Proceedings of the 6th ACM Conference on Recommender Systems, RecSys'12, pp. 115–122. ACM (2012)
Li, S., Zha, Z.J., Ming, Z., Wang, M., Chua, T.S., Guo, J., Xu, W.: Product comparison using comparative relations. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'11, pp. 1151–1152. ACM (2011)
Lippert, C., Weber, S.H., Huang, Y., Tresp, V., Schubert, M., Kriegel, H.P.: Relation prediction in multi-relational domains using matrix factorization. In: Proceedings of the NIPS 2008 Workshop 'Structured Input–Structured Output' (2008)
Liu, B.: Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers, San Rafael (2012)
Liu, B., Zhang, L.: In: In: Aggarwal, C., Zhai, C. (eds.). Mining Text Data, (ed.) A survey of opinion mining and sentiment analysis, pp. 415–463. Springer (2012)
Liu, K., Xu, L., Zhao, J.: Opinion target extraction using word-based translation model. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL '12, pp. 1346–1356. ACL (2012)
Liu, H., He, J., Wang, T., Song, W., Du, X.: Combining user preferences and user opinions for accurate recommendation. Electron. Commer. Res. Appl. **12**(1), 14–23 (2013)
McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM Conference on Recommender Systems, RecSys'13, pp. 165–172. ACM (2013)
McAuley, J., Yang, A.: Addressing complex and subjective product-related queries with customer reviews. In: Proceedings of the 25th International Conference on World Wide Web, WWW'16, International World Wide Web Conferences Steering Committee, pp. 625–635 (2016)
McAuley, J., Leskovec, J., Jurafsky, D.: Learning attitudes and attributes from multi-aspect reviews. In: Proceedings of the 12th IEEE International Conference on Data Mining, ICDM'12, pp. 1020–1025. IEEE (2012)
McCallum, A.K.: Mallet: a machine learning for language toolkit (2002). http://mallet.cs.umass.edu. Accessed 15 Dec 2017
Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
Moshfeghi, Y., Piwowarski, B., Jose, J.M.: Handling data sparsity in collaborative filtering using emotion and semantic based features. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'11, pp. 625–634. ACM (2011)
Musto, C., Semeraro, G., Polignano, M.: A comparison of lexicon-based approaches for sentiment analysis of microblog posts. In: Proceedings of the 8th International Workshop on Information Filtering and Retrieval, DART'14, pp. 59–68 (2014)
Musto, C., de Gemmis, M., Semeraro, G., Lops, P.: A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews. In: Proceedings of the 11th ACM Conference on Recommender Systems, RecSys'17, pp. 321–325. ACM (2017)
Nie, Y., Liu, Y., Yu, X.: Weighted aspect-based collaborative filtering. In: Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'14, pp. 1071–1074. ACM (2014)
Ning, X., Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 37–76. Springer (2015)
Pazzani, M.J.: A framework for collaborative, content-based and demographic filtering. Artif. Intell. Rev. **13**(5–6), 393–408 (1999)
Pero, Š., Horváth, T.: Opinion-driven matrix factorization for rating prediction. In: Proceedings of the 21st International Conference on User Modeling, Adaptation, and Personalization, UMAP'13, pp. 1–13. Springer (2013)
Pilászy, I., Zibriczky, D., Tikk, D.: Fast als-based matrix factorization for explicit and implicit feedback datasets. In: Proceedings of the 4th ACM Conference on Recommender Systems, RecSys'10, pp. 71–78. ACM (2010)

Poirier, D., Tellier, I., Fessant, F., Schluth, J.: Towards text-based recommendations. In: Proceedings of the 9th International Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information, RIAO'10, pp. 136–137 (2010)

Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of the 2007 Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT'05, pp. 9–28. Springer (2005)

Poria, S., Cambria, E., Ku, L.W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. In: Proceedings of the 2nd Workshop on Natural Language Processing for Social Media, SocialNLP'14, pp. 28–37 (2014)

Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Comput. Linguist. **37**(1), 9–27 (2011)

Raghavan, S., Gunasekar, S., Ghosh, J.: Review quality aware collaborative filtering. In: Proceedings of the 6th ACM Conference on Recommender Systems, RecSys'12, pp. 123–130. ACM (2012)

Rana, T.A., Cheah, Y.N.: Aspect extraction in sentiment analysis: comparative analysis and survey. Artif. Intell. Rev. **46**(4), 459–483 (2016)

Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, AUAI'09, pp. 452–461. AUAI Press (2009)

Salakhutdinov, R.R., Mnih, A.: Probabilistic matrix factorization. In: Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS'07, pp. 1257–1264 (2007)

Scaffidi, C., Bierhoff, K., Chang, E., Felker, M., Ng, H., Jin, C.: Red opal: product-feature scoring from reviews. In: Proceedings of the 8th ACM Conference on Electronic Commerce, EC'07, pp. 182–191. ACM (2007)

Schuster, S., Manning, C.D.: Enhanced English universal dependencies: an improved representation for natural language understanding tasks. In: Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC'16 (2016)

Seroussi, Y., Bohnert, F., Zukerman, I.: Personalised rating prediction for new users using latent factor models. In: Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia, HT'11, pp. 47–56. ACM (2011)

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP'13, pp. 1631–1642 (2013)

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Comput. Linguist. **37**(2), 267–307 (2011)

Titov, I., McDonald, R.: Modeling online reviews with multi-grain topic models. In: Proceedings of the 17th International Conference on World Wide Web, WWW'08, pp. 111–120. ACM (2008a)

Titov, I., McDonald, R.T.: A joint model of text and aspect ratings for sentiment summarization. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, ACL'08, vol. 1, pp. 308–316. ACL (2008b)

Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, NAACL-HLT'03, vol. 1, pp. 173–180. ACL (2003)

Van den Oord, A., Dieleman, S., Schrauwen, B.: Deep content-based music recommendation. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS'13, vol. 2, pp. 2643–2651 (2013)

Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'11, pp. 448–456. ACM (2011)

Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: a rating regression approach. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'10, pp. 783–792. ACM (2010)

Wang, Y., Liu, Y., Yu, X.: Collaborative filtering with aspect-based opinion mining: a tensor factorization approach. In: Proceedings of the 12th IEEE International Conference on Data Mining, ICDM'12, pp. 1152–1157. IEEE (2012)

Wang, F., Pan, W., Chen, L.: Recommendation for new users with partial preferences by integrating product reviews with static specifications. In: Proceedings of the 21st Conference on User Modeling, Adaptation, and Personalization, UMAP'13, pp. 281–288. Springer (2013)
Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the 3rd Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT-EMNLP'05, pp. 347–354. ACL (2005)
Wu, Y., Ester, M.: Flame: a probabilistic model combining aspect based opinion mining and collaborative filtering. In: Proceedings of the 8th ACM International Conference on Web Search and Data Mining, WSDM'15, pp. 199–208. ACM (2015)
Wu, C.Y., Diao, Q., Qiu, M., Jiang, J., Wang, C.: Jointly modeling aspects, ratings and sentiments for movie recommendation. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'14, pp. 193–202. ACM (2014)
Yates, A., Joseph, J., Popescu, A.M., Cohn, A.D., Sillick, N.: Shopsmart: product recommendations through technical specifications and user reviews. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM'08, pp. 1501–1502. ACM (2008)
Zhang, L., Liu, B., Lim, S.H., O'Brien-Strain, E.: Extracting and ranking product features in opinion documents. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10, pp. 1462–1470. ACL (2010)
Zhang, W., Ding, G., Chen, L., Li, C., Zhang, C.: Generating virtual ratings from chinese reviews to augment online recommendations. ACM Trans. Intell. Syst. Technol. **4**(1), 9 (2013)
Zhang, S., Yao, L., Sun, A.: Deep learning based recommender system: a survey and new perspectives (2017). arXiv preprint arXiv:1707.07435
Zhao, W.X., Jiang, J., Yan, H., Li, X.: Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP'10, pp. 56–65. ACL (2010a)
Zhou, T., Kuscsik, Z., Liu, J.G., Medo, M., Wakeling, J.R., Zhang, Y.C.: Solving the apparent diversity-accuracy dilemma of recommender systems. Proc. Natl. Acad. Sci. **107**(10), 4511–4515 (2010b)

María Hernández-Rubio is a Senior Data Scientist at BBVA Data and Analytics, the second largest commercial bank in Spain. She collaborates with the Information Retrieval group at Universidad Autónoma de Madrid, where she received a M.Sc. in Computational Intelligence in 2017, and researched on recommender systems exploiting user textual reviews. At industry, she has focused on Machine Learning (ML) and building ML-production systems for the banking industry, working on a wide range of topics, such as Smart Cities, Data for Social Good, and Marketing. Currently, her main research interests are in NLP-based solutions to improve the relationships between customers and bank managers.

Iván Cantador is a Senior Lecturer of Computer Science at Universidad Autónoma de Madrid (UAM), Spain. He received a Ph.D. degree in Computer Science in 2008 at UAM. After earning his Ph.D., he worked as a research associate at University of Glasgow, UK, and as a postdoctoral research visitor at the Free University of Bolzano (Italy). His current research interests fall in the areas of Recommender Systems, Personalized Information Retrieval, and User Modeling. Dr. Cantador has published over 70 conference and journal papers, and serves regularly as reviewer and scientific committee member in venues of the aforementioned fields, including prestigious conferences such as RecSys, SIGIR, ECIR, WWW, CIKM, Hypertext and UMAP, to name a few.

Alejandro Bellogín is an Assistant Professor at Universidad Autónoma de Madrid (UAM), Spain. Previously, he worked with Prof. Arjen de Vries associated to the Centrum Wiskunde and Informatica under a post-doctoral Marie Curie research Grant. In 2012, he completed his Ph.D. under the supervision of Prof. Pablo Castells and Dr. Iván Cantador at UAM. Dr. Bellogín has worked in several areas of user modeling and personalization, including recommender systems, evaluation, and reproducibility, where he has published over 50 conference and journal papers, while being involved in venues related to these areas, such as RecSys, WWW and UMAP, among others.