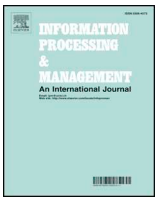
Information Processing and Management 57 (2020) 102304

Contents lists available at ScienceDirect

Information Processing and Management

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



What makes a review a reliable rating in recommender systems?

Dionisis Margaris[a], Costas Vassilakis[b,*], Dimitris Spiliotopoulos[b]

[a] Department of Informatics and Telecommunications, University of Athens, Greece
[b] Department of Informatics and Telecommunications, University of the Peloponnese, Greece

ARTICLE INFO

Keywords:
Recommender systems
Collaborative filtering
Rating prediction using textual information
Confidence level
Feature selection

ABSTRACT

The way that users provide feedback on items regarding their satisfaction varies among systems: in some systems, only explicit ratings can be entered; in other systems textual reviews are accepted; and in some systems, both feedback types are accommodated. Recommender systems can readily exploit explicit ratings in the rating prediction and recommendation formulation process, however textual reviews -which in the context of many social networks are in abundance and significantly outnumber numeric ratings- need to be converted to numeric ratings. While numerous approaches exist that calculate a user's rating based on the respective textual review, all such approaches may introduce errors, in the sense that the process of rating calculation based on textual reviews involves an uncertainty level, due to the characteristics of the human language, and therefore the calculated ratings may not accurately reflect the actual ratings that the corresponding user would enter. In this work (1) we examine the features of textual reviews, which affect the reliability of the review-to-rating conversion procedure, (2) we compute a confidence level for each rating, which reflects the uncertainty level for each conversion process, (3) we exploit this metric both in the users' similarity computation and in the prediction formulation phases in recommender systems, by presenting a novel rating prediction algorithm and (4) we validate the accuracy of the presented algorithm in terms of (i) rating prediction accuracy, using widely-used recommender systems datasets and (ii) recommendations generated for social network user satisfaction and precision, where textual reviews are abundant.

1. Introduction

Collaborative filtering (CF) creates personalized recommendations by taking into account ratings expressed by users (Koren & Bell, 2011). CF algorithms initially identify people having similar tastes, by examining the likeness of already entered ratings (Margaris & Vassilakis, 2017a; Zhou & Han, 2019); for each user U, other users having highly similar preferences with U are designated as U's nearest neighbors (NNs). Subsequently, in order to predict the rating that U would give to an item I that he has not reviewed yet, the ratings assigned to item I by U's NNs are combined (Ahmadian, Meghdadi, & Afsharchi, 2018; Lee & Brusilovsky, 2017), under the assumption that if users have exhibited similar tastes in the past, they are highly likely to do so in the future as well (Karimi, Jannach, & Jugovac, 2018; Margaris & Vassilakis, 2017a; Sánchez & Bellogín, 2019). CF has been proven to be the most successful and popular approach for building recommender systems (RS) (Najafabadi, Mohamed, & Onn, 2019). The Pearson correlation coefficient (see also Section 3) is a widely accepted metric to measure correlation between users in CF-based RSs (Camacho &

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

* Corresponding author at: Department of Informatics and Telecommunications, University of the Peloponnese, Akadimaikou G. K. Vlachou, 22100, Greece.

E-mail addresses: margaris@di.uoa.gr (D. Margaris), costas@uop.gr (C. Vassilakis), dspiliot@uop.gr (D. Spiliotopoulos).

https://doi.org/10.1016/j.ipm.2020.102304
Received 12 October 2019; Received in revised form 10 May 2020; Accepted 13 May 2020

Alves-Souza, 2018; Margaris & Vassilakis, 2018).

However, typical CF algorithms can only work with numeric user ratings. Many systems, including Amazon, TripAdvisor, IMDB and so forth, require that user feedback on items includes an explicit numeric rating, and in these cases the user's feedback can be directly used by CF-based recommender systems. With the advent of social media however, user feedback on items may be shifted to unstructured, textual-only formats which cannot be directly handled by recommender systems. For instance, many users publish their opinions on locations, products or companies through tweets, in which case only textual information is provided. On Facebook, while the facility to enter numeric reviews on pages is available, Facebook users are reluctant to use it and mainly resort to expressing their opinions through posting of textual content: for instance, the Denver Art Museum has gathered approximately 4K explicitly entered user ratings from Facebook users, however the number of textual posts on the same page is significantly higher, accounting for 225,000 check-ins, and a number of additional (non-check-in) posts which is approximated to 130,000. Clearly, the potential to convert posts to numerical ratings would substantially increase the density of the user-item rating matrix (the sum of check-ins and non-check in posts is 355,000, i.e. 88 times higher than the number of explicitly entered user ratings), allowing the CF-based system to compute personalized rating prediction for more users, and increase recommendation accuracy and usefulness. Under this perspective, the accuracy of the textual review to numeric ratings conversion procedure is crucial. Nevertheless, due to the characteristics of the human language, the process of numeric rating calculation based on textual reviews involves an uncertainty level, hence no review-to-rating system is error-proof.

In this work we firstly investigate the characteristics of user reviews which affect the review-to-rating conversion procedure. We also propose and validate an algorithm, which initially computes the confidence level for each individual review-to-rating conversion and attaches this factor to the review; subsequently, this confidence level is exploited both in the users' similarity computation phase and in the prediction formulation phase of CF, in order to produce more reliable predictions. Results show that the proposed algorithm achieves very satisfactory results in (i) the CF datasets tested, improving rating prediction accuracy, when users enter not only ratings but also reviews in a RS database and (ii) the extent to which users are satisfied from the recommendations that the proposed algorithm formulates, which has been shown to be higher than the case where the confidence factor is not used.

An alternative approach to using textual reviews in recommender systems is to directly build user profiles from the textual reviews, without converting textual reviews to numeric ratings, and subsequently generating recommendations based on these profiles and the items' textual reviews. A survey of such methods is presented in Chen, Chen, & Wang, 2015. While some of these methods show promising results, which in some cases exceed the performance of predictions based on explicitly entered star ratings -such as the cases of Ganu, Kakodkar, & Marian, 2013 and Wang, Liu, & Yu, 2012- it is worth noting that the achieved rating prediction quality is owing among others to the exploitation of additional features (also termed as *aspects*) extracted from the textual reviews, such as the movie plots and genres or restaurant atmosphere food, service and price. These aspects have been shown to contribute significantly to prediction accuracy: for instance, Frémal & Lecron, 2017 report that genre-based clustering can improve the prediction RMSE between 4.7% and 9.8%. The algorithm presented in this paper focuses on the trustworthiness aspect of the reviews, without examining or exploiting such additional data. It is possible however that the rating prediction algorithm presented in this paper, which takes into account the confidence level computed on the basis of textual review features, can be extended to consider genre-based data, as in the work by Frémal & Lecron, 2017, or other item aspect data. The relevant genre/aspect data can either be extracted from the user reviews, or be retrieved from other sources, e.g. the IMDB dataset includes numerous aspect attributes including the movie genre(s), the director, leading actors and so forth. It is also noteworthy that the algorithm proposed in this paper can be applied in conjunction with other algorithms that target at either the improvement of prediction accuracy in CF systems by (i) taking into account data sourced from social networks (SN) (Camacho & Alves-Souza, 2018; Margaris, Vassilakis, & Georgiadis, 2018; Margaris & Vassilakis, 2016; Marinho et al., 2011), (ii) considering shifts in marking practices (Li, Shan, Jheng, & Chou, 2016; Margaris & Vassilakis, 2018, Margaris & Vassilakis, 2017a) and (iii) removing old user ratings (Margaris & Vassilakis, 2017, Margaris & Vassilakis, 2016), or methods for accelerating prediction computation, such as clustering (J. Chen, Uliji, Wang, & Yan, 2018; Margaris, Georgiadis, & Vassilakis, 2015; Xu, 2018).

1.1. Research questions

In this work we investigate the characteristics of user reviews that affect the review-to-rating conversion procedure. We also propose and validate an algorithm that initially computes the confidence level for each individual review-to-rating conversion and attaches this factor to the review. Subsequently, this confidence level is exploited both in the user similarity computation phase and in the prediction formulation phase of CF, to produce more reliable predictions. The paper addresses the following research questions:

RQ1: when converting textual reviews to numeric ratings, does there exist some association between features of textual reviews and the reliability of numeric ratings that have been produced from these reviews?
RQ2: if an association does exist, which features exhibit the strongest association?
RQ3: how can the reliability of a numeric rating that has been produced on the basis of a textual review be quantified, considering the linguistic features of the textual review?
RQ4: how can the review reliability scores be taken into account by a rating prediction algorithm to support the generation of more successful recommendations?

1.2. Paper contributions

Considering the above, our paper extends the state-of-the-art in RSs through:

(1) exploring features of textual reviews that can be exploited for the computation of the reliability levels for numerical ratings, that are calculated based on textual reviews, and assessing the effectiveness of each feature. A new feature is identified in this context, which significantly surpasses the performance of features identified in previous works,
(2) developing an algorithm for computing the reliability level of each textual review, considering the associated features,
(3) enhancing standard CF algorithms with the potential to take into account reliability levels in the rating prediction process and
(4) evaluating the performance of the proposed algorithm, considering both (i) SNs that do not support SN relationships among users (e.g. IMDB and Amazon) i.e. cases in which the SN essentially directed towards the collection, organization and sharing of user-contributed content (Obar & Wildman, 2015; Ureña, Kou, Dong, Chiclana, & Herrera-Viedma, 2019) and (ii) SNs that do support such relationships (e.g. Facebook and Twitter).

The current work has a number of theoretical implications. Firstly, to the best of our knowledge, the reliability of numeric ratings that have been computed on the basis of textual reviews has only been addressed in a recent publication by Margaris, Vassilakis, & Spiliotopoulos, 2019 in the context of a social network recommender. This work advances the research presented in Margaris, Vassilakis, & Spiliotopoulos, 2019 as follows:

1) we introduce the use of a new textual feature that is proven to be considerably more effective; using this feature, user reviews are found to produce up to 40% more reliable ratings than others, on average (the respective number of the feature that was proposed by Margaris, Vassilakis, & Spiliotopoulos, 2019 was 12%).
2) we thoroughly investigate the procedure of textual feature extraction and rating prediction quality assessment, enabling other researchers to gain insight on the methodology employed and conduct new research towards the identification of prominent textual features of reviews to be exploited in the computation of reliability and methods for quantifying reliability levels.
3) the algorithm proposed in this paper can be applied to any item domain, while the algorithm proposed by Margaris, Vassilakis, & Spiliotopoulos, 2019 is specific to venue recommendation.
4) the algorithm proposed in this paper may operate in two types of SNs, (i) SNs that do not support SN relationships among users and (ii) SNs that do support such relationships, while the algorithm proposed in Margaris, Vassilakis, & Spiliotopoulos, 2019 requires and exploits information about social ties among users, being thus applicable only to the second type of SNs.
5) we provide a more thorough evaluation of the proposed approach, which is based on both objective measurements (prediction error metrics against the ground truth established by widely used RS datasets) and subjective opinions (a user survey); contrary, the evaluation of the approach presented in Margaris, Vassilakis, & Spiliotopoulos, 2019 uses only subjective means, since it targeted the improvement of recommendation formulation quality in SNs.
6) the proposed approach is evaluated by using both RS datasets and SN users; the technique proposed in Margaris, Vassilakis, & Spiliotopoulos, 2019 was evaluated only by the latter, since its target was to upgrade only venue recommendation formulation quality in SNs.
7) we disassociate the computation of reliability from the social network recommendation method, allowing thus the application of the presented methods both in the RS domain, as well as in other domains, such as opinion mining and reputation management. Additionally, new algorithms that consider the reliability levels in the rating prediction and/or recommendation formulation process can be devised.

In terms of practical implications, the algorithms proposed in this work are shown to be both effective and efficient. To this end, they can be directly incorporated into existing RSs, to improve rating prediction accuracy and leverage user satisfaction from recommendation quality, as well as recommendation precision.

The rest of the paper is structured as follows: Section 2 overviews related work, while Section 3 presents the proposed algorithm for incorporating confidence levels in the CF-based rating prediction process. Section 4 explores textual features, assessing their suitability as predictors of the textual review-to-rating mapping quality, and Section 5 evaluates the proposed algorithm in terms of (i) rating prediction accuracy using seven contemporary CF datasets, (ii) SN user satisfaction regarding the offered recommendations, as well as recommendation precision and (iii) overhead, due to the computation of the confidence level for each individual review-to-rating conversion. Finally, Section 6 concludes the paper and outlines future work.

2. Related work

In the last decade, there has been an intense research activity in RSs, and several algorithms for formulating recommendations have been proposed. CF computes personalized recommendations by firstly identifying people having similar likings with the user u for whom the recommendation is generated (these users are termed as the *near neighbors of u*), and then taking into consideration ratings expressed by *u's* near neighbors (Desrosiers & Karypis, 2011). The similarity of likings is quantified by analyzing the closeness of ratings that have already been recorded (Zhou & Han, 2019). The CF-based approach is the most promising recommendation technique used in RSs to make predictions based on liked-minded user preferences (Najafabadi, Mohamed, & Onn, 2019).

In order to improve recommendation accuracy, knowledge from other sources can be exploited, such as Knowledge-based Systems

(Margaris, Vassilakis, & Georgiadis, 2017; Vijayakumar, Vairavasundaram, Logesh, & Sivapathi, 2019), Internet of Things (IoT) (Hassani, Haghighi, Ling, Jayaraman, & Zaslavsky, 2018; Margaris & Vassilakis, 2017), SNs (Camacho & Alves-Souza, 2018; Margaris, Vassilakis, & Georgiadis, 2017; Xu, 2018), Information Retrieval Systems (Bellogín & Said, 2019; Valcarce, Bellogín, Parapar, & Castells, 2018) and Neural Networks (Paradarami, Bastian, & Wightman, 2017; Ying et al., 2018).

Furthermore, information from item reviews can be utilized for the generation of recommendations. Qian, Zhang, Ma, Yu, & Peng (2019) propose a hybrid information fusion-based RS that also considers emotions conveyed by reviews, in order to address the challenge of improving the quality of recommender services, in which three representative types of information are fused to comprehensively analyze the user's features: (a) explicit information, deriving from user rating data, (b) implicit information, deriving from SN data and (c) emotional information, deriving from user reviews.

Ngo-Ye, Sinha, & Sen (2017) target at predicting the online customer reviews' helpfulness by employing a text regression model augmented with cognitive scripts. Cognitive scripts represent shared knowledge in a domain, and in the context of this work cognitive scripts are created by assigning to human annotators the task of highlighting phrases that, in the annotators' view, are important for determining review helpfulness. Subsequently, the highlighted phrases are compiled into a script lexicon for a given domain, which in this respect constitutes the shared conception of essential elements, which determine the review helpfulness' evaluation. The words in the entries of the script lexicon are utilized as features in a text regression model to predict review helpfulness. Furthermore, Ngo-Ye, Sinha, & Sen (2017) develop and empirically validate a novel approach for combining script analysis and dimension reduction and finally present a new method to (a) predict review helpfulness and (b) evaluate the efficiency and effectiveness of the scripts-enriched model.

A multi-text summarization technique for identifying the top-k most informative sentences of hotel reviews is proposed by Hu, Chen, & Chou (2017). They consider factors like author credibility and conflicting opinions and develop a new sentence importance metric. Both the sentiment similarities and content (Lops, de Gemmis, & Semeraro, 2011) are used to determine the similarity between two sentences. In order to identify the top-k sentences, the k-medoids clustering algorithm is used to partition the sentences into *k* groups, where the medoids from the aforementioned groups are selected as the final summarization results. Zheng, Noroozi, & Yu (2017) present the so-called Deep Cooperative Neural Networks (DeepCoNN), which is a deep model that examines review texts to jointly learn user behavior as well as item properties. DeepCoNN consists of two parallel neural networks (NNs) coupled in the last layer. The first NN focuses on learning user behavior, by analyzing reviews written by the user, while the second one learns item properties from the reviews written for it. Finally, the last layer that couples the NNs together, receives inputs from both parallel NNs, enabling the interaction between latent factors learned for users (sourced from the first NN) and items (sourced from the second NN), following the factorization machine techniques paradigm.

Nilashi et al. (2018) develop a new recommendation method for formulating hotel recommendations in e-tourism platforms, by using multi-criteria ratings. Furthermore, they use both unsupervised and supervised machine learning techniques in order to analyze the customers' online reviews. Musto, de Gemmis, Semeraro, & Lops (2017) propose a multi-criteria RS, based on CF techniques, that processes user reviews to extract the information expressed therein, and then utilizes this information to compile a multi-faceted representation of users' interests. To identify the salient features that should be included in the user profiles, the authors exploit an opinion mining and sentiment analysis framework which automatically extracts relevant aspects and sentiment scores from users' reviews. Finally, the authors create and propose a multi-criteria CF algorithm, which predicts user interest in particular items and produces recommendations, based on users' multi-faceted representations.

Bauman, Liu, & Tuzhilin (2017) propose a recommendation technique that recommends items of interest to the user, enhanced with the recommendation of specific aspects of consumption of the items -especially pertaining to aspects that the user can control- to further enhance the user experience with those items. For instance, the recommendation of a specific restaurant (item) to a user can be augmented with the recommendation for ordering some specific foods (e.g. "seafood") there (an aspect of consumption that the user has control on) or the time to visit the restaurant (e.g. breakfast vs launch). The generation of enhanced recommendations is accomplished by applying sentiment analysis on the user reviews: firstly, the sentiment that the user may have towards the item is predicted by considering what the user might express about the item's different aspects, and subsequently the most important aspects of the user's potential experience with that item.

Jadidinejad, Macdonald, & Ounis (2019) introduce a weak supervision approach, which is applied at the stage of data pre-processing to unify both implicit and explicit feedback datasets, targeting to bridge the gap between the tasks of rating prediction and ranking. The proposed approach accomplishes to address the underrepresentation of items with low popularity in the formulated recommendations, achieving thus a lower popularity bias (Jadidinejad, Macdonald, & Ounis, 2019) while in parallel achieving improved rating prediction accuracy and either improved or at least comparable ranking prediction accuracy, as contrasted to other methods.

In the absence of explicit ratings entered by users, many works exist that create ratings based on textual information, Abhishek, Divyashree, Keerthana, Meghana, & Karthik (2019) propose a classification model for predicting the star rating of a given review. The necessary features are extracted using either a n-gram model or a bag of words model in order to build the feature vectors. The aforementioned feature vectors are then combined with supervised learning algorithms for constructing a classification model. Finally, as far as the classification model logistic regression is concerned, naïve Bayes and support vector algorithms have proven to perform with better accuracy. Seo, Huang, Yang, & Liu (2017) create models of item properties and user preferences by employing an attention-based convolutional NN (CNN). The CNN encompasses two distinct subnetworks, the item network and the user network. The item (user) network is fed with item (user) documents to extract latent representations of items (users). Finally, the outputs of the two networks are combined under a dot product operator to join the latent factors and predict rating values for the user-item pairs. Since the two subnetworks are combined into a single one, the training phase is also performed jointly, enabling the interaction

between the items' and users' latent factors, in a fashion similar to the approach introduced by Zheng, Noroozi, & Yu (2017). Both subnetworks include an attention layer, which selects the most informative words for user and item profiles, while a visualization tool is also provided, which exploits the output of the attention layers to deliver insight on the words that are selected as the most representative of the items' properties and user preferences.

Yi, Huang, & Qin (2018) aim to ameliorate the performance of recommendation formulation on top of sparse datasets, by processing reviews and ratings to extract the latent factors of users and items. In this context, adversarial learning techniques are employed to regularize the distributions of the latent factors of users and items, so that both distributions are better aligned. Experiments reported in this work, show that the alignment of distributions results in increased rating prediction accuracy and recommendation quality.

Cieslik (2019) computes ratings from texts using word embeddings combined with a Long short-term memory neural network with two layers. Several other implementations for producing ratings from text also exist, such as the Yelp Rating and Review Trends (Lester, 2019), Yelp Star Rating (Bathula, 2019; Logesh, Subramaniyaswamy, Vijayakumar, & Li, 2019), Vader Sentiment (Hutto, 2019), RecSys BrickMover's code (BrickMover team, 2019), used in many works (Amin, Hossain, Akther, & Alam, 2019; Kronmueller, Chang, Hu, & Desoky, 2018; Qiu, Liu, Li, & Lin, 2018), however for most of these works, no details on the techniques used are currently available. Using the above works, ratings can be produced from reviews; hence standard CF algorithms can then be applied to recommend items to users.

However, these approaches do not consider the factor of reliability in the process of review-to-rating conversion, and therefore this reliability is neither computed and explicitly represented; nor taken into account in the recommendation process. Furthermore, the issue of investigating the characteristics that make a user review more reliable to produce a rating by a review-to-rating system is not addressed in the current literature. Recently, Margaris, Vassilakis, & Spiliotopoulos (2019) have proposed a method for generating venue recommendations for SN users, considering qualitative parameters of the venues (e.g. service, atmosphere, price), the habits of users concerning these qualitative parameters, semantic and physical distance of venues as well as a collaborative filtering score. The collaborative filtering score encompasses the strength of SN relationships and the likeness of users' tastes, with the latter being computed on the basis of both explicitly entered numerical venue scores and scores calculated by processing textual reviews. While this work considers the reliability in the conversion between textual reviews and numerical scores, it provides little information on the evaluation of the effect that textual features have on the reliability of the computed numeric scores, as well as on the method for determining the mapping between textual review features and conversion confidence levels. Moreover, the evaluation presented in Margaris, Vassilakis, & Spiliotopoulos (2019) report SN user satisfaction levels from an algorithm encompassing numerous dimensions (venue QoS, semantic distance, physical distance, SN tie strength, conversion reliability levels), thus the effect of the inclusion of the reliability factor cannot be adequately isolated and evaluated. It has to be noted here that the work in Margaris, Vassilakis, & Spiliotopoulos (2019) explore three textual review features (document length, total number of positive and negative terms within the document and absolute difference between the number of positive and negative terms within the document), while in this work we introduce a new textual feature, namely *polarity term density*, which achieves significantly better performance than the *absolute difference between the number of positive and negative terms within the document* feature, which was the one shown to exhibit the best performance among the features explored in Margaris, Vassilakis, & Spiliotopoulos (2019). Finally, in this work we assess the efficiency of the computation of reliability levels for textual reviews and evaluate the gains of the proposed approach both in the context of datasets with no social relations between users and in SNs where such relations are established.

3. Introducing rating confidence level in the rating prediction process

CF-based systems predict the rating that a user U would assign to an item i by first identifying a set of users who have rated items similarly with U in the past; this set of users is termed as "*U's* near neighbors" (NNs). A number of metrics have been proposed to quantify the similarity between the ratings of two users, with the Pearson correlation coefficient (Camacho & Alves-Souza, 2018; Pereira, Plastino, Zadrozny, & Merschmann, 2018) being the most widely used similarity metric in CF-based RSs. According to the Pearson correlation coefficient, the similarity between two users U and V is expressed as:

$$sim(U,\ V) = \frac{\sum_k (r_{U,k} - \overline{r_U})*(r_{V,k} - \overline{r_V})}{\sqrt{\sum_k (r_{U,k} - \overline{r_U})^2}* \sqrt{\sum_k (r_{V,k} - \overline{r_V})^2}} \quad (1)$$

where k iterates over items for which both U and V have already entered ratings, while $\overline{r_U}$ and $\overline{r_V}$ denote the mean value or ratings entered by users U and V, respectively. Then, for user U, the users exhibiting the highest similarity values with U are designated as *U's* near neighbors, denoted as NN_U. Subsequently, in order to predict the rating that U would assign to an item i (which U has not rated in the past), the ratings given by *U's* NNs to that item are combined as shown in formula (2):

$$p_{U,i} = \overline{r_u} + \frac{\sum_{V \in NN_u} sim(U, V)*(r_{V,i} - \overline{r_V})}{\sum_{V \in NN_u} sim(U, V)} \quad (2)$$

The proposed algorithm modifies the prediction computation phase, by taking into account the confidence level associated with each individual rating. More specifically, formula (1) is modified as follows:

$$sim(U,\ V) = \frac{\sum_k (r_{U,k} - \overline{r_U})*CL_{U,k}*(r_{V,k} - \overline{r_V})*CL_{V,k}}{\sqrt{\sum_k ((r_{U,k} - \overline{r_U})*CL_{U,k})^2} * \sqrt{\sum_k ((r_{V,k} - \overline{r_V})*CL_{V,k})^2}} \quad (3)$$

and formula 2 is modified as shown in equation 4:

$$p_{U,i} = \overline{r_u} + \frac{\sum_{V \in NN_u} sim(U, V)*(r_{V,i} - \overline{r_V})*CL_{V,i}}{\sum_{V \in NN_u} sim(U, V)*CL_{V,i}} \quad (4)$$

where $CL_{x,y}$ is the confidence level assigned to the rating of user x on item y; the value of the confidence level depends on its provenance, i.e. whether it was explicitly entered or computed based on reviews; in the former case, the value of $CL_{x,y}$ equals 1.0, while in the latter case, the value of $CL_{x,y}$ is less than or equal to 1.0, and depends on the features of the review text. In Eqs. 3 and 4 $\overline{r_U}$ and $\overline{r_V}$ denote the weighted average of the corresponding user's ratings, where weighting is based on the confidence level assigned to each individual rating. Formally, the weighted average is computed as shown in equation 5:

$$\overline{r_u} = \frac{\sum_i CL_{U,i}*r_{U,i}}{\sum_i CL_{U,i}} \quad (5)$$

Regarding the features taken into account to calculate the confidence level for ratings computed from textual reviews, in this work, we consider the following four ones:

- *Document Length (DL)*, i.e. the number of words within the review, under the assumption that a lengthier document provides more information about the user's opinion.
- *Polarity Term Count (PTC)*, computed as the total count of terms expressing polarity (either positive or negative) within the review, under the rationale that terms expressing polarity convey stronger evidence about the user's opinion. The list of terms expressing polarity is drawn from the opinion lexicon (Liu, 2019; Xing, Pallucchini, & Cambria, 2019). Note that in this case we do not identify and manage the presence of negations, because only the number of polarity terms is significant, regardless of whether they express a positive or negative view. For example, the review text “The movie was good” has PTC = 1, while the same is true for the review text “The movie was not good”, which includes a negation.
- *Document polarity level (DPL)*, computed as the absolute value of the difference between the positive and negative term count within the review, under the assumption that if positive terms significantly outnumber negative ones (or vice versa), the review polarity level is stronger. The list of terms expressing polarity (positive or negative) is again drawn from the opinion lexicon (Liu, 2019; Xing, Pallucchini, & Cambria, 2019). In this case, we need to identify and manage the presence of negation that is associated with polarity terms, since positive and negative terms are not uniformly treated in the document score computation. Negation handling is performed using the method introduced by (Pang, Lee, & Vaithyanathan, 2002) and further elaborated on by Chikersal, Poria, Cambria, Gelbukh, & Siong (2015): this method asserts that nouns, adjectives, adverbs, or verbs that are located between specific negation words and the immediately following punctuation marks are also negated. The set of negation words for which this approach is used is drawn from Chikersal, Poria, Cambria, Gelbukh, & Siong (2015). For example, the review text “The main course was good and the desert was delicious” has two terms with positive polarity, having thus DPL = 2, while in the review text “The main course was not good, but the desert was delicious” one of the terms with positive polarity (“good”) is negated under the presence of “not” and counted thus as a term with negative polarity. Having one term with positive polarity and one with a negative one, leads to a DPL score equal to 0. More elaborate methods for determining the document polarity can be applied, including the handling of negations that do not invert but rather neutralize the basic term polarity (e.g. “not bad” may convey a neutral opinion instead of a positive one), as well as the handling of modals neutralizing polarity terms (e.g. “If acting was better we would be more satisfied”, where “better” and “satisfied” have positive valence but are neutralized by the existence of “if”); the examination of such methods will be a part of our future work.
- *Polarity term density (PTD)*, which is computed as the ratio of the document polarity level (i.e. the absolute value of the difference between the positive and negative terms count within the review), to the review length, under the rationale that a high polarity level, combined with high occurrence frequency of terms expressing polarity conveys stronger evidence about the user's opinion.

The first three features were also explored in Margaris, Vassilakis, & Spiliotopoulos (2019) (with the second and third feature being referenced therein as total number of positive and negative terms within the document (TNPNT) and absolute difference between the number of positive and negative terms within the document (ANPNT)), and are included here for self-containment purposes. The fourth feature, *Polarity term density (PTD)*, is a novel concept introduced and explored in this work and is shown to achieve significantly higher performance than the features considered in Margaris, Vassilakis, & Spiliotopoulos (2019).

The algorithm presented above is the first step towards the investigation of RQ4 that is further discussed and concluded in Section 5.3.

4. Exploring review features

In this section we investigate whether the features of textual reviews listed in Section 3 can be positively associated with an increase of rating prediction accuracy, so as to address RQ1 and RQ2.

We explored the existence of such associations experimentally, testing each feature F individually, according to the experimental

protocol described below:

(1) for each test dataset D, we selected the user ratings r_k that contained *both* a textual review $t(r_k)$ and a numeric score $n(r_k)$; the set of the selected ratings will be denoted as D'. Then, each textual review was converted to a numerical rating $n^c(r_k)$, and subsequently we calculated the precision of the conversion procedure, as this is expressed by the Mean Absolute Error (MAE) metric, taking into account that for each computed numerical rating $n^c(r_k)$, the corresponding explicitly entered numeric score $n(r_k)$ constitutes the ground truth. Formally, the MAE for the conversion procedure is computed as:

$$MAE = \frac{1}{n} \sum_{r_k \in D'} |n^c(r_k) - n(r_k)| \tag{6}$$

(2) Afterwards, for each test dataset D' we introduced a threshold Th_F for the considered feature F; the value of the threshold Th_F was then increased in an iterative fashion and within each iteration we retained only the reviews in D' for which the value of the considered feature F was greater than or equal to the current value of Th_F. For instance, when the *Polarity Term Count* feature was considered and the threshold Th_{PTC} was set to 2, only the ratings whose textual reviews included two or more polarity terms were retained. Formally, the set of retained user reviews of D' is expressed as

$$D'_F(Th_F) = \{r_k \in D' : f(r_k) \geq Th_F\} \tag{7}$$

where $f(r_k)$ denotes the value of feature F for review r_k. Finally, we converted the textual reviews in $D'_F(Th_F)$ to numeric scores and calculated the MAE of the conversion procedure as described in step (1), above, and expressed in Eq. (6).

Note that the filtering of test datasets according to the Th_F threshold is performed at this phase to explore the existence of positive associations between the different review features and increments of rating prediction accuracy. In the normal operation of the algorithm, where textual reviews are converted to numeric ratings that are inserted into the user-item rating matrix, all textual reviews are considered without the application of any filter or threshold.

This association discovery experimental procedure is based on the following rationale: if elevated values for a textual feature (e.g. polarity term count) are positively correlated with higher numeric score prediction accuracy, then retaining in the dataset only reviews having high values for the feature under consideration (for instance, reviews with a high number of polarity terms) should result in the calculation of more accurate computed numeric scores and, consequently, the MAE would decrease. Concerning the four textual features described in the previous section, the respective threshold values used in the experimental procedure are as follows:

- For the *Document Length (DL)* feature, threshold values from 10 up to 100 were examined, with the increment step being equal to 10.
- Regarding the *Polarity Term Count (PTC)* feature, in the first iteration only reviews having two or more polarity terms were considered (i.e. threshold = 2); subsequently threshold values of 5, 10, 15, 20, 25, 30, 35, 40 and 50 were used.
- For the *Document polarity level (DPL)* feature, in the first iteration only reviews having a polarity level greater than or equal to 1 were considered (i.e. threshold = 2); subsequently threshold values of 2, 3, 4, 5, 6, 8, 10, 15 and 20 were used.
- Regarding the *Polarity term density (PTD)* feature, the threshold values from 2% up to 20% were examined, with the increment step being equal to 2%.

In this experimental exploration, we used datasets containing explicit ratings along with their comments, for every rating; these datasets provide a "ground truth" for every user input (the explicitly entered rating), against which the rating computed on the basis of the textual review can be compared. Such datasets are available in the Amazon Dataset compilation (McAuley, Pandey, & Leskovec, 2015; McAuley, Targett, Shi, & van den Hengel, 2015); the Yelp Dataset[1] also covers the aforementioned requirements. Thus, effectively, the sets of ratings D and D' referenced in steps (1) and (2) above coincide, however the procedure described in steps (1) and (2) can be also applied to datasets where only some of the user ratings contain both a textual review and a numeric score. Furthermore, the datasets used in our experiments have the following properties:

(1) they are up to date, while at the same time containing reviews spanning over two decades, and are widely used for benchmarking in CF research and
(2) they vary with respect to the type of dataset item domain (electronics, movies, music, books, beauty, videogames and restaurants), size (from 54MB to 9.4GB, in JSON format) as well as number of users, items and reviews.

Table 1 summarizes the basic properties of these datasets.

Fig. 1 depicts the results obtained from the experiment described above; for the conversion of textual reviews to ratings, the Yelp Review Classifier (Tran, 2019) was used, and for each feature the average MAE for all datasets is presented for conciseness purposes. The MAE metrics observed for individual datasets have small deviations from the presented mean, and generally follow the same

[1] https://www.yelp.com/dataset.

Table 1
Datasets Summary.

Dataset Name	#users	#reviews	dataset size (in JSON format)
Amazon “Videogames"[a]	22K	230K	304MB
Amazon “CDs and Vinyl"2	75K	1.1M	1.3GB
Amazon “Movies and TV"2	120K	1.7M	1.9GB
Amazon “Books"2	350K	8.9M	9.4 GB
Amazon “Electronics"2	190K	1.7M	1.4GB
Amazon “Office Products"2	5K	53K	54MB
Amazon “Beauty"2	22K	200K	134MB
Yelp challenge[b]	6M	6M	4.5GB

[a] More information on the datasets is provided in McAuley, Pandey, & Leskovec, 2015; McAuley, Targett, Shi, & van den Hengel, 2015
[b] More information on the datasets is provided in Hutto (2019)

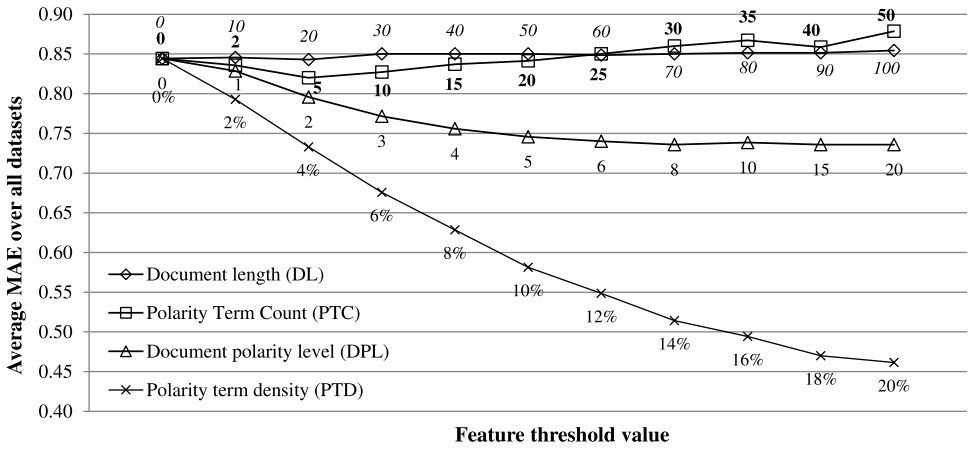


Fig. 1. Effect of review features in the accuracy of rating prediction when using the Yelp Review Classifier.

pattern. The per-dataset behavior regarding the PTD feature is discussed further below and illustrated in Fig. 2, while a respective analysis of the per-dataset behavior for the DL, PTC and DPL features is available in Margaris, Vassilakis, & Spiliotopoulos (2019). In Fig. 1 we can observe that the review length and the polarity term count features are not associated with an increment in rating prediction accuracy, since confining the experiment dataset to lengthier reviews or reviews with larger number of positive and negative terms, does not lead to a reduction in the MAE. Considering the DPL feature, when the respective threshold increases, a MAE drop is observed, which increases along with the threshold, up to the point that the threshold reaches the value of 8: from that point

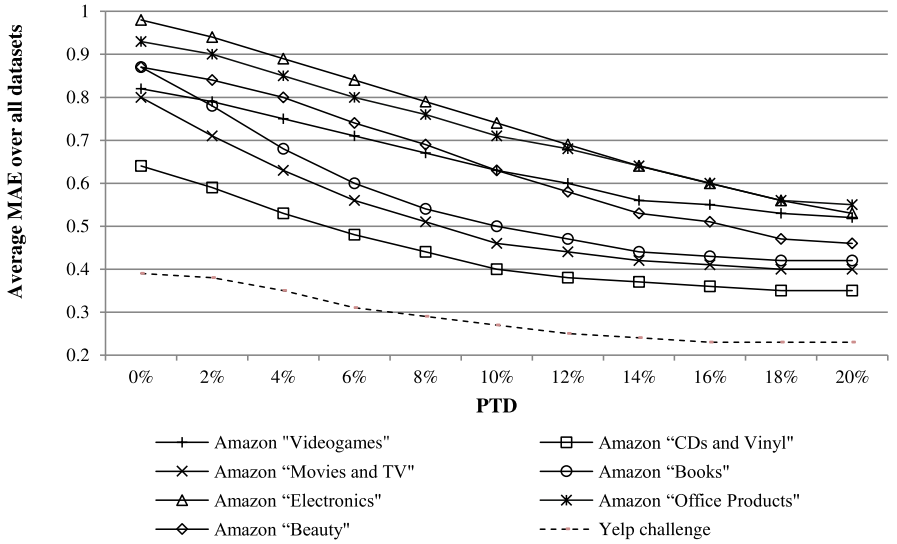


Fig. 2. Error reduction on the individual dataset level, when using the Yelp Review Classifier in combination with the PTD feature.

onwards, the MAE reduction remains practically constant, equal to 12.85%. Finally, we can observe that the PTD feature is very strongly associated with a MAE reduction: when the PTD threshold increases, the MAE drops, with the MAE reduction being approximately linear up to the point that the threshold reaches the value of 12%; beyond that point, the MAE still declines as the threshold value increases, albeit with a smaller rate. The overall MAE reduction achieved when the PTD feature threshold increases from 0% to 20% is 44.3%.

This experiment provides strong indications that the PTD feature can achieve the best results regarding the improvement of rating prediction accuracy, surpassing the performance of the DPL feature introduced by Margaris, Vassilakis, & Georgiadis (2018), by a performance margin up to 31.45%.

Fig. 2 depicts the results obtained from the experiment described above (Yelp Review Classifier), for all datasets, for the PTD feature that proved to be the optimal feature based on the relation between the prediction error and the PTD feature value.

We can clearly see that similarly to the behavior observed in Fig. 1, the prediction error drops more sharply when the PTD value increases up to the value of 12%. The maximum prediction error reductions are observed for the Amazon "Books" dataset and the Amazon "Movies &TV" dataset, with the prediction error reduction reaching 51.7% and 50%, respectively, when the PTD = 20%, while the minimum prediction error reductions are observed for the Amazon "Videogames" dataset and the Amazon "Electronics" dataset, with the prediction error reduction reaching 36.6% and 40.9%, respectively, when the PTD = 20%. It has to be mentioned that in all the datasets tested the prediction error reduction when using the PTD feature is at least 53% larger than the reduction achieved when using the DPL feature, which has proven to be the second best feature (e.g. as far as the Amazon "Videogames" dataset is concerned, which achieved the lowest prediction error reduction -36.6%- the respective prediction error reduction, when using the DPL feature – which proved to yield the second best results in the previous experiment – is only 11%).

To verify that the use of the particular reviews to ratings conversion tool (Yelp Review Classifier) did not introduce a bias in the results, we repeated the same experiment using a second conversion tool, namely VADER Sentiment Analysis (Hutto, 2019; Zhang, Zhang, Chan, & Rosso, 2019). The results obtained are depicted in Fig. 3. Similar to the case of Fig. 1, only the average MAE reduction across all datasets is presented: in this experiment too, the MAE metrics observed for individual datasets have small deviations from the presented mean, and generally follow the same pattern.

In this figure we can observe that, when using the VADER Sentiment Analysis tool, again neither the DL feature, nor the PTC feature can be positively associated with a reduction in the MAE. When the DPL feature is considered, the MAE drops up to the point that the DPL reaches the value of 5, while beyond that point no further reductions in the MAE are obtained. The maximum MAE reduction achieved is 8.1%. Finally, we can observe that when the PTD increases up to the value of 18%, the MAE drops almost linearly, while from that point and onwards the improvement rate declines. The experiment conducted with the PTD threshold set to 20% yields a MAE improvement of 31.9%, as compared to the baseline measurement (i.e. the experiment where the full dataset is used). Again, we observe that the PTD feature achieves the best results: it surpasses the performance of the DPL feature introduced by Margaris, Vassilakis, & Georgiadis (2018), by a performance margin up to 23.8%.

Additionally, by comparing Fig. 1 and Fig. 3, we can conclude that despite the differences in the absolute magnitudes, the MAE reduction patterns in relation to the four features considered are very similar for both tools. In general, the Yelp review classifier has shown to provide more accurate conversions from textual reviews to numerical ratings than the VADER Sentiment Analysis tool, incurring however increased computational costs (computation time increases between 20% and 30%).

Fig. 4 depicts the results obtained from the experiment described above (VADER Sentiment Analysis Tool), for all datasets, for the PTD feature that proved to be the optimal feature based on the relation between the prediction error and the PTD feature value. We can clearly see that, similarly to the behavior observed in Fig. 3, the prediction error drops almost linearly when the PTD value increases up to the value of 18%. The maximum prediction error reductions are observed for the Amazon "Videogames" dataset and the Amazon "Movies and TV" dataset, with the prediction error reduction reaching 36.4% and 36.3%, respectively, when the PTD = 20%. The minimum prediction error reductions are observed for the Amazon "Office Products" dataset and the Amazon "Beauty" dataset, with the prediction error reduction reaching 22.1% and 26.4%, respectively, when the PTD = 20%. It has to be

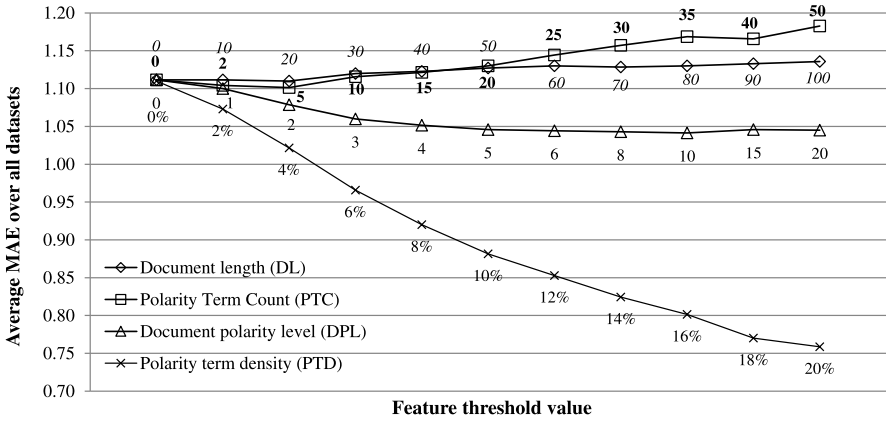


Fig. 3. Effect of review features in the accuracy of rating prediction when using the VADER Sentiment Analysis tool.

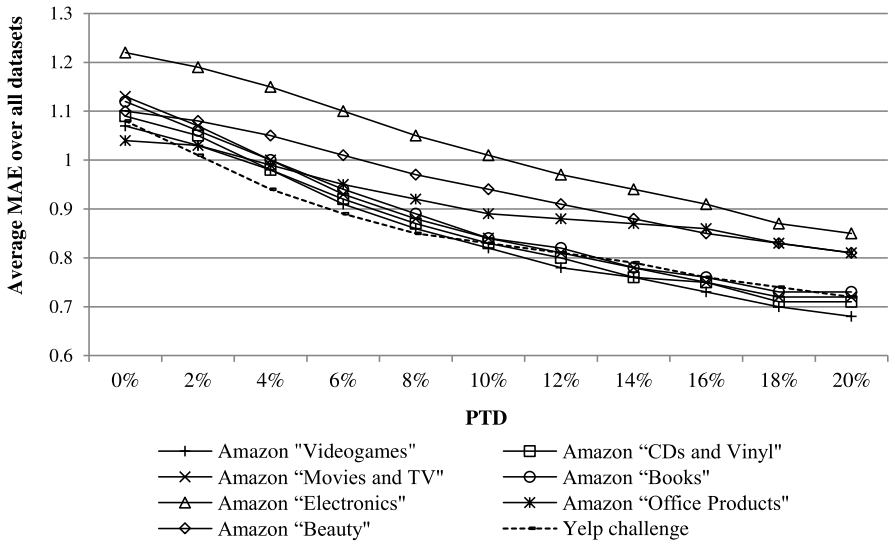


Fig. 4. Error reduction at individual dataset level, when using the VADER Sentiment Analysis tool in combination with the PTD feature.

mentioned that, in all datasets tested, the prediction error reduction when using the PTD feature value is at least 225% higher than the reduction achieved when using the DPL feature, which has proven to be the second best feature (e.g. as far as the Amazon "Office Products" dataset is concerned, which achieved the lowest prediction error reduction (22.1%), the respective prediction error reduction when using the DPL feature –which proved to yield the second best results in the previous experiment–is only 8.57%).

Taking all the above into account, it is clear that the PTD feature is the one found to be most strongly associated with rating calculation accuracy, surpassing the performance of the DPL feature introduced in (Margaris, Vassilakis, & Spiliotopoulos, 2019) by up to 31.45% (when the Yelp Review Classifier is used) or by up to 23.8% (when the VADER sentiment analyzer is used).

Before concluding with the choice of the PTD feature, we explored the probability density function for the PTD feature, to ascertain that a sufficient number of reviews are available for the different ranges of the PTD feature values that affect the MAE, as shown in the above experiments. For this exploration we used the following datasets:

(a) the datasets listed in Table 1.
(b) the OpinRank Data[2] (Ganesan & Zhai, 2012). This dataset contains approximately 300K reviews (40K car reviews and 260K hotel reviews).
(c) the Google Local Reviews[3] dataset (He, Kang, & McAuley, 2017; Pasricha & McAuley, 2018). This dataset contains 11.5M reviews from 4.5M users.

Datasets (b) and (c) were chosen to supplement the datasets listed in Table 1 because (i) they constitute real-world datasets widely listed in CF research (ii) pertain to different domains than those listed in Table 1 and (c) have different review text/linguistic characteristics than the datasets listed in Table 1, as analyzed below. Therefore, the extended dataset selection offers increases the confidence level to the conclusion that real-world datasets do contain a sufficient number of reviews are available for the different ranges of the PTD feature values that affect the MAE.

Datasets (b) and (c) were preprocessed to remove reviews containing characters from non-Latin alphabets, since the opinion words lexicons used in the experiments contained only English language words. Still, the filtered review set did not contain only reviews in the English language (e.g. reviews in French, Spanish or Italian were not removed and the respective review texts are not bound to contain any English language opinion terms, introducing a skew towards low PTD values), however the objective of the experiment was not to compute precisely the probability density function for the PTD feature, but rather gain an overall insight towards the trends that this function expresses.

Fig. 5 illustrates the results of the probability density function exploration experiment. We can observe that every PTD range has at least 5% of the total reviews. The distribution of PTD values is not uniform across all datasets: the Amazon, Yelp and OpinRank datasets follow roughly the same pattern, with each of the PTD ranges [0%, 5%) and [5%, 10%) accounting for approximately the 30% of the reviews in the dataset, and this percentage declining for the subsequent PTD ranges. On the other hand, the Google Local Reviews dataset exhibits a lower density in the first two ranges, coupled with a significantly higher density for the PTD range [20%, 100%]. This is attributed to the fact that the Google local reviews dataset contains a considerable amount of short-length/strong opinion review texts, e.g. "Great", "Wonderful", "Sucks", "Very good food, great service, my home away from home! Great quality!",

[2] available at https://kavita-ganesan.com/entity-ranking-data.
[3] available at https://cseweb.ucsd.edu/~jmcauley/datasets.html#google_local.

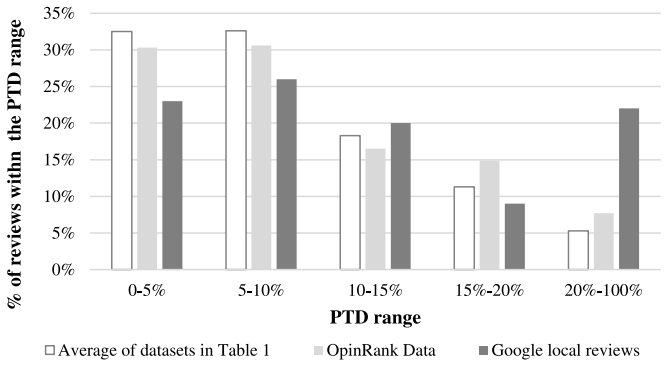


Fig. 5. Probability distributions for the PTD feature.

"Lamest service ever", while the other datasets contain in general more lengthy and descriptive reviews, where opinion words are scarcer. Overall, the results from this experiment affirm that a sufficient number of reviews are available for the different ranges of the PTD feature values that affect the MAE.

In view of all the above findings, in the remaining part of our paper we finalize the choice of the PTD feature, and we will consider only this feature in the rest of this paper.

5. Experimental evaluation

In this section, we report on our experiments aiming to:

(1) determine the optimal CL value for the numeric scores computed from textual reviews, considering the feature selected in the previous section and
(2) evaluate the performance of the proposed approach, in terms of (a) prediction accuracy, (b) SN user satisfaction and precision, regarding the recommendations they are offered; the recommendations offered to SN users are formulated based on the rating predictions generated by the proposed algorithm and (c) overhead introduced by the computation of the confidence level for each individual review-to-rating conversion.

Our experiments were run on a machine equipped with six Intel Xeon E7 4830 @ 2.13GHz CPUs, 256GB of RAM and one 900GB HDD with a transfer rate of 200MBps, which hosted the datasets and ran the rating prediction algorithms. In these experiments, we used the datasets described in Section 4 above and summarized in Table 1.

5.1. Determining the optimal CL value for computed ratings

As noted above, the first goal of our experiments is to determine the optimal CL value for the numeric scores computed from textual reviews, considering the PTD feature introduced in this work. Effectively, we aim to compute an optimal function *CLev(ptd)*: [0..1] → [0..1] to map the PTD value of each textual review $tr_{U,i}$ to a confidence level $CL_{U,i}$, which will then be used in formulas 3-5 listed in Section 3. This part addresses RQ3 ("how can the reliability of a numeric rating that has been produced on the basis of a textual review be quantified, considering the linguistic features of the textual review?).

To this end, we examined two mapping function schemes:

(1) in the first scheme, the domain of the textual review's PTD values [0, 100%] was partitioned into quantization intervals, and each quantization interval was then mapped to a constant confidence level. In this context, different partitioning settings were explored, with the number of quantization intervals varying from 5 to 10, while additionally a number of different settings for the location of quantization interval endpoints were tested. The quantization interval endpoint values considered in our experiments were 0%, 2.5%, 5%, 7.5%, 10%, 12.5%, 15%, 17.5%, 20%, 22.5% and 100%. Interval endpoint values higher than 22.5% were not tested since, according to the experiments presented in Section 4, only minimal MAE improvements are harvested when the value of PTD increases beyond the point of 20%, indicating thus that the confidence levels to the computed numerical rating values should remain stable when PTD increases beyond the point of 20%. Henceforth, splitting the interval [22.5%, 100%] into two subintervals [22.5%, x) and [x, 100%] with 22.5% x < 100%, would be unavailing because, according to the discussion above, both subintervals would map to the same confidence level value, being therefore equivalent to a single interval mapping to the same value.

For each such partitioning setting, multiple experiments were run, with each experiment examining a different set of mappings between quantization intervals within the partitioning scheme and confidence level *CLev* values. Following the findings of the experiment described in Section 4, quantization intervals corresponding to smaller PTD values were mapped to lower *CLev* values.

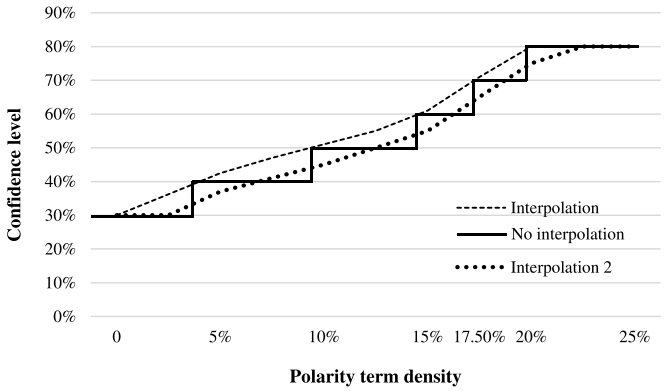


Fig. 6. Non-interpolating versus interpolating functions.

For example, formula 8 below depicts a quantization scheme employing the following six quantization intervals: [0, 5%), [5%, 10%), [10%, 15%), [15%, 17.5%), [17.5%, 20%), [20%, 100%], mapping each interval to a constant value.

$$CLev_c(ptd) = \begin{cases} 0.3, & if\ 0\% \le ptd < 5\% \\ 0.4, & if\ 5\% \le ptd < 10\% \\ 0.5, & if\ 10\% \le ptd < 15\% \\ 0.6, & if\ 15\% \le ptd < 17.5\% \\ 0.7, & if\ 17.5\% \le ptd < 20\% \\ 0.8, & if\ 20\% \le ptd \le 100\% \end{cases} \quad (8)$$

(1) the second scheme again explored different quantization intervals as in option (1), above, however the PTD values in each interval were mapped to confidence levels using an interpolation function, to account for variations in PTD values within each quantization interval. In our experiments we used the standard linear interpolation function:

$$\mathrm{CLev_i(ptd)} = \frac{(\mathrm{ptd} - \mathrm{q_l(ptd)})*(\mathrm{CLev_c(q_h(ptd))} - \mathrm{CLev_c(q_l(ptd))})}{\mathrm{q_h(ptd)} - \mathrm{q_l(ptd)}} + \mathrm{CLev_c(q_l(ptd))} \quad (9)$$

where $q_l(ptd)$ and $q_h(ptd)$ are the left and right endpoints (i.e. lower and upper bound), respectively, of the quantization interval that *ptd* belongs to.

Fig. 6 charts the non-interpolating function expressed in formula (8) (the line tagged as "no interpolation"), and its interpolating counterpart (the line tagged as "interpolation"). In this figure we can observe that $CLev_i(ptd) \ge CLev_q(ptd)$, $\forall ptd \in [0\%, 100\%]$, and this fact may introduce a skew. However, in Fig. 6 we can also notice the line tagged as "Interpolation 2", which follows more closely the values of the non-interpolating function "no interpolation". The line "Interpolation 2" corresponds to the partitioning setting using 7 intervals ([0, 2.5%), [2.5%, 7.5%), [7.5%, 12.5%), [12.5%, 15%), [15%, 20%), [20%, 22.5%), [22.5%, 100%]), following the interpolated scheme, which was also considered in the exploration for the optimal arrangement of optimal function *CLev(ptd)*. Generalizing, for every non-interpolating function $CLev_{NI}(ptd)$ following the first scheme, a corresponding interpolating function $CLev_I(ptd)$ was tested which was closely observing the values of $CLev_{NI}(ptd)$.

In the previous step, a set of potential *CLev(ptd)* mapping functions *MF* was created. In order to identify the optimal one, i.e. the one that minimizes the MAE, we run a set of experiments which proceeded as follows:

(1) For each dataset *D*, we created mixtures *M* of explicitly entered ratings, and numeric scores computed from textual reviews, using the Yelp Review Classifier. In this experiment we used only the Yelp Review Classifier for converting textual reviews to ratings, since -according to the results of the experiments presented in Section 4- it performs considerably better than the VADER sentiment analyzer. The ratio of explicitly entered ratings to the total number of ratings ranged from 10% to 90%, proceeding at 10% increments. Explicitly entered ratings were tagged with a confidence level of 100%, while numeric scores computed from textual reviews were tagged with the confidence level computed by the mapping function. The mixture corresponding to dataset D, having a percentage of explicitly entered ratings equal to *pct* is denoted as D_{pct}. The rationale behind the creation of these mixtures is to assess how well each mapping function performs under different ratios of explicitly entered ratings to the total number of ratings.

(2) For each mixture D_{pct} and each mapping function $mf \in MF$, we executed an experiment $E^{mf}_{pct}(D)$, which consisted of the application of a CF procedure using the dataset D_{pct}, in the context of which the values of PTD for the computed ratings were mapped to confidence levels using the mapping function *mf*. Within this experiment, the MAE was computed by applying a 5-fold cross validation (James et al., 2017), with the 80% of the dataset being used as a training set and the remaining 20% of the dataset being used as a test set. Note that in this process, the Yelp dataset was excluded, since each of its users had rated only one item,

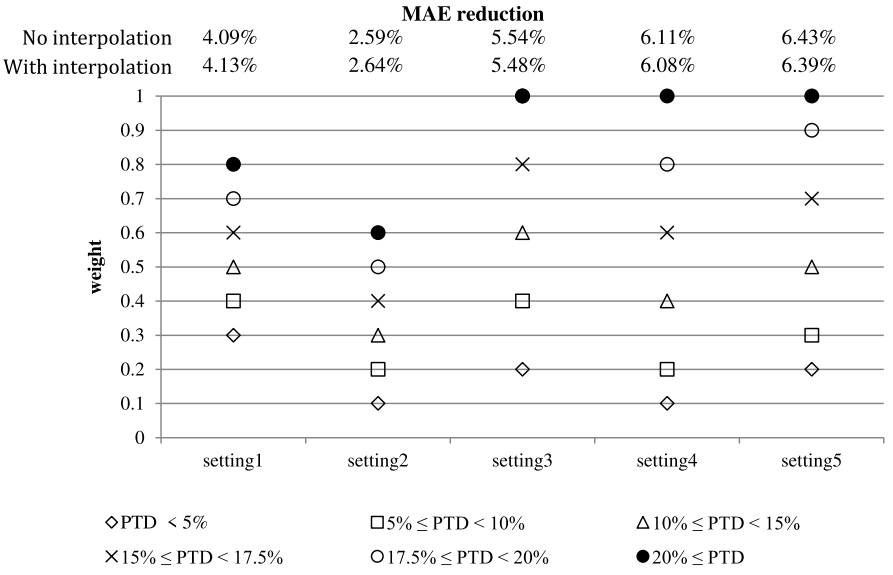


Fig. 7. Effect of weight assignment and use of interpolation on the accuracy of rating prediction.

hence CF cannot be applied.

(3) Finally, for each mapping function $mf \in MF$, we computed the average normalized MAE improvement $\overline{MAE(mf)}$ for that mapping function considering all experiments $E_{pct}^{mf}(D)$ involving function *mf*. The baseline for measuring improvement was the performance of the standard CF algorithm, in which confidence levels are not taken into account. Formally,

$$\overline{\mathrm{MAE(mf)}} = \frac{1}{N} \sum_{D_{pct} \in mixtures} \frac{\mathrm{MAE^{plainCF}}(D_{pct}) - \mathrm{MAE}(E_{pct}^{mf}(D))}{\mathrm{MAE^{plainCF}}(D_{pct})} \tag{10}$$

where N is the number of experiments involving the mapping function *mf*, $MAE(E_{pct}^{mf}(D))$ is the MAE for experiment $E_{pct}^{mf}(D)$ and $MAE^{plainCF}(D_{pct})$ is the MAE produced by the plain CF algorithm for mixture D_{pct}. Note that we use the normalized improvement against the baseline to provide a fair comparison basis, amortizing the effect of varying absolute magnitudes of MAE across different mixtures.

Fig. 7 depicts the MAE reduction achieved by five non-interpolating mapping functions as well as the MAE reduction achieved when using their interpolating counterparts. All these functions employ the following quantization intervals: [0, 5%), [5%, 10%), [10%, 15%), [15%, 17.5%), [17.5%, 20%) and [20%, 100%]; the non-interpolating variant of setting 5 depicted in Fig. 7 was found to achieve the greatest MAE reduction among all mapping functions tested[4]. Along with setting 5, which achieves the optimal result, we present the results of four additional mapping functions (for both their non-interpolating and interpolating variants), to provide insight on the effect of the mapping functions on the overall MAE reduction. Regarding the presence of interpolation, we can observe that in all cases its effect is small and can be positive or negative. For setting 5, in particular, the presence of interpolation leads to a slightly inferior performance, hence the non-interpolating variant is chosen.

Recapitulating, the optimal mapping function *CLev* for the computation of the confidence level, has been found to be:

$$CLev_c(ptd) = \begin{cases} 0.2, & if\ 0\% \le ptd < 5\% \\ 0.3, & if\ 5\% \le ptd < 10\% \\ 0.5, & if\ 10\% \le ptd < 15\% \\ 0.7, & if\ 15\% \le ptd < 17.5\% \\ 0.9, & if\ 17.5\% \le ptd < 20\% \\ 1, & if\ 20\% \le ptd \le 100\% \end{cases} \tag{11}$$

5.2. Rating prediction accuracy

In this subsection, we elaborate on the performance of the proposed algorithm, examining how the algorithm performs in specific CF datasets, as well as its performance in cases where the ratio of the number of numeric scores computed from textual reviews to the

[4] Mapping functions that are equivalent to the one represented by setting 5 depicted in Fig. 7 but employing finer granularities (e.g. [0, 2.5%) → 0.3, [2.5%, 5) → 0.3 . . .) clearly achieve the same MAE reduction; here, we present only one of the family of equivalent mapping functions.

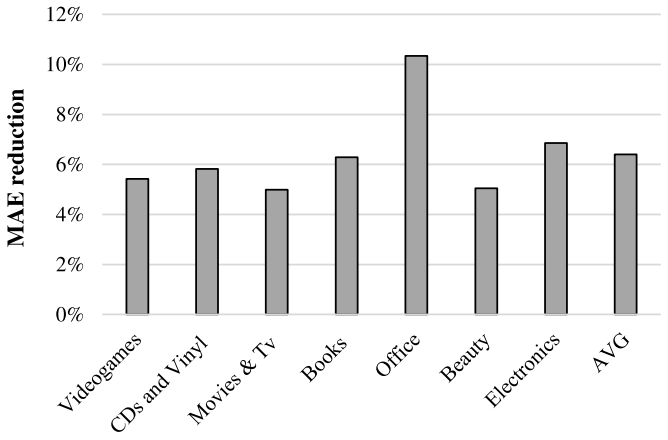


Fig. 8. MAE reduction achieved by considering the CL weights to the ratings produced by user reviews in various percentage mixtures.

total number of ratings is high.

Fig. 8 illustrates the performance of the proposed algorithm for each of the examined datasets, under the optimal mapping function determined in Section 5.1 (Eq. 11). The value depicted in Fig. 8 for each dataset corresponds to the average MAE reduction achieved for the specific dataset, considering all ratios of explicitly entered ratings to the total number of ratings (recall that this ratio ranged from 10% to 90%, under increments of 10%). In all cases, the baseline for measuring improvement was the performance of the standard CF algorithm, in which confidence levels are not taken into account.

In Fig. 8 we can observe that the MAE improvement ranges from 5.0% in the Movies & TV dataset and Beauty dataset, to 10.3% in the Office dataset, with an average of 6.4%. This variation in performance is attributed to the particular features of the textual reviews in the different datasets. More specifically, the “Office" and the “Electronics" datasets -in which the highest improvements are observed- are the ones where positive and negative terms are attributed with the highest probability to the user's perception on the quality of the item, since the description of the corresponding items is highly sentiment-neutral. On the contrary, in other datasets user reviews are bound to contain positive and negative terms which refer to the product itself (e.g. the term “beauty" is counted a positive polarity term, however in the review "This is another beauty product" in the “Beauty" dataset it is actually a neutral one). Similarly, a textual review “This made me cry. It's a great sad song" is considered to have one negative term (“cry") and one positive one (“great sad song") while actually both terms are positive in the particular context. Deeper analysis of these aspects is needed, which is planned in our future work.

Fig. 9 focuses on the case that the ratio of number of numeric scores computed from textual reviews to the total number of ratings is high. More specifically, Fig. 9 depicts the MAE improvement for the case where the mixture on which the CF experiment was performed consisted of 10% explicitly entered ratings and 90% ratings that had been computed on the basis of textual reviews. Again, the plain CF algorithm, which does not take into account the review confidence levels, was used as a yardstick.

In Fig. 9 we can observe that the average MAE improvement achieved due to the introduction and use of confidence levels across all datasets is 8.9%. This improvement ranges from 7.4% in the Movies & TV dataset, to 12.9% in the Office dataset. Comparing with the corresponding results of Fig. 8, we can notice that the MAE improvement is considerably higher when the mixture contains a high ratio of ratings that have been computed on the basis of textual reviews (38.5% on average). This indicates the ability of the proposed

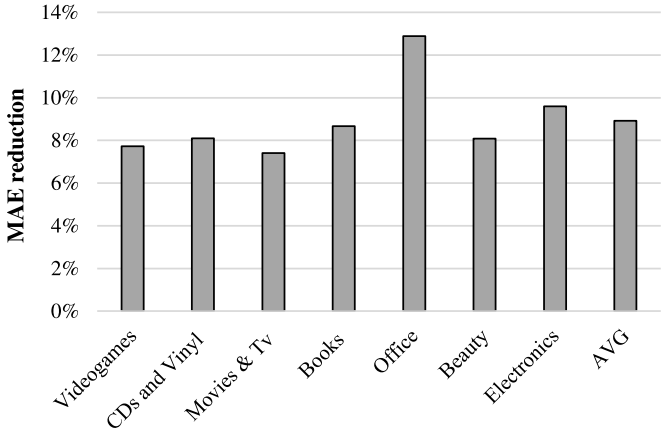


Fig. 9. MAE reduction achieved by considering the CL weights to the ratings produced by user reviews, when 90% of the ratings were computed on the basis of textual reviews.

algorithm to operate under context with elevated levels of reliability and successfully deal with the inaccuracies introduced to ratings during the textual review-to-rating conversion process, while on the other hand the performance of the plain CF algorithm, which does not take into account confidence levels, is considerably degraded in such contexts.

5.3. Social network user satisfaction and recommendation precision

After having determined the optimal mapping function *CLev* for the computation of the confidence level and validated the efficiency of the proposed algorithm towards producing more accurate predictions, we conducted an additional experiment, which aimed to quantify (i) the recommendation precision and (ii) the extent to which users were satisfied from the recommendations they received, when the algorithms presented in Sections 3 and 4 are used as a basis for rating prediction; this satisfaction level was compared to that obtained from other related algorithms. This analysis contributes towards the investigation of RQ4.

The experiment evaluated user satisfaction and precision regarding offered recommendations considering two distinct cases of SN:

1. a SN where no direct relationships among users are established and the SN is essentially directed towards the collection, organization and sharing of user-contributed content (Obar & Wildman, 2015; Ureña, Kou, Dong, Chiclana, & Herrera-Viedma, 2019); typical examples of such SNs are IMDB and Amazon, and
2. a SN where direct relationships between users can be established (Obar & Wildman, 2015; Ureña, Kou, Dong, Chiclana, & Herrera-Viedma, 2019), and these relationships are subsequently exploited by the recommendation generation algorithm; typical examples of such SNs are Facebook and Twitter.

In order to evaluate the quality of recommendations, we carried out a user survey in which 50 subjects participated. The subjects were staff and students from the community of the University of Athens, Greece, and were selected from four diverse academic departments (theater studies, physics, medicine and computer science). The users' mean age was 28 years, with a minimum of 18 years and a maximum of 51. All participants had been registered Facebook users for 4 years or more, using it regularly (one hour per day or more, and at least for 6 days per week). Each user had posted a number of reviews or check-ins (which were complete with textual data) to Facebook; the number of reviews and check-ins ranged from 63 to 281 with a mean of 105. The minimum number of Facebook friends among the participants was 73 and the maximum was 632, with a mean of 229. The profile and review/check-in data required by the algorithms were extracted using the Facebook Graph API[5].

In order to measure and demonstrate the gains introduced by the proposed algorithm, we have considered the following four recommendation generation algorithms:

1. *A plain CF algorithm without user relationship information* (*plain CF, no rel*): numeric ratings are produced from textual reviews based only on the Yelp rating prediction (no confidence level is used). Then, a standard CF rating prediction algorithm is applied (c.f. Eq. 2), and the items attaining the top-*K* rating predictions constitute the recommendation to the user.
2. *A confidence level-enhanced algorithm without user relationship information (CL-enhanced, no rel)*. The proposed algorithm is used to generate rating predictions, computing and exploiting the rating confidence level. Once rating predictions have been computed, the items attaining the top-*K* rating predictions constitute the recommendation to the user.
3. *A plain CF algorithm exploiting user relationship information* (*plain CF, with rel*): similarly to case (1), numeric ratings are produced from textual reviews based only on the Yelp rating prediction (no confidence level is used). However, in the rating prediction process, the *tie strength* (Bakshy, Eckles, Yan, & Rosenn, 2012; Valverde-Rebaza, Roche, Poncelet, & de Andrade Lopes, 2018) between users moderates the degree to which each user's opinion is taken into account in the computation of the rating prediction. As shown in Bakshy, Eckles, Yan, & Rosenn (2012), SN users respond considerably better to recommendations (such as proposals and adverts) that stem from SN friends to which the user is closely related. The "closeness" of the relation is quantified via the *tie strength* measure; the tie strength between two users u_1 and u_2 is computed on the basis of the number of interactions between u_1 and u_2 in the recent past. Again, the top-*K* rating predictions constitute the recommendation to the user.
4. *A confidence level-enhanced algorithm exploiting user relationship information (CL-enhanced, with rel)*: this is similar to case (3), however a confidence level is computed when a textual review is converted into a rating, and this confidence level is exploited in the rating prediction process.

The specific procedures for tie strength computation and rating prediction computation followed in the algorithms in cases (3) and (4) are given in (Margaris, Vassilakis, & Spiliotopoulos, 2020). . These two cases correspond to a SN where direct relationships between users can be created, while cases (1) and (2) simulate a SN where no direct relationships between users can be established. The same report also lists the detailed results of the social network user satisfaction experiment.

In more detail, in the context of the experiment each participant was requested to assign scores to 20 item recommendations suggested to her, on a scale from 1 (very unsatisfactory) to 10 (very satisfactory). Each of the recommendation generation algorithms (1) to (4) presented above contributed five recommendations. The order in which recommendations were presented to the users for evaluation was randomized. If an item was recommended by more than one algorithm, then the item was listed only once in the list of

[5] https://developers.facebook.com/docs/graph-api.

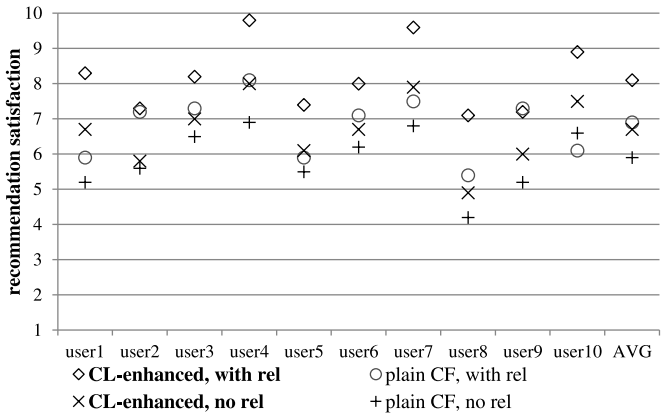


Fig. 10. User satisfaction feedback regarding the recommendations presented

recommendations offered to the user, and the score given for that item was accounted to all proposing algorithms. The recommended items were drawn from the "CDs and Vinyl", "Movies and TV" and "Electronics" domains.

Fig. 10 depicts the satisfaction of participants from the recommendations generated for them, on a scale of 1 to 10, for the algorithms mentioned above. On average (last column in Fig. 10), the proposed algorithm employing the textual-review-to-rating confidence level (computed using the polarity term density feature) and exploiting the SN user relationships attains an overall user satisfaction of 8.1, outperforming all other approaches. In particular, when this approach is compared with the *"plain CF, with rel"* algorithm, i.e. its counterpart that does not use the confidence level, which achieves an average user satisfaction of 6.9, an improvement of 17.4% is observed. Similarly, in SN contexts where no relationships among users are established, the "*CL-enhanced, no rel"* which uses the textual-review-to-rating confidence level outperforms the *"plain CF, no rel"* algorithm (which is its counterpart that does not use the confidence level) by a margin of 13.6%.

Within Fig. 10 we have also included the results regarding ten individual users that have been chosen to demonstrate that the performance of the algorithm is not uniform across all cases. In 92% of the cases (46 out of 50 users) the *"CL-enhanced, with rel"* algorithm was ranked higher than *"plain CF, with rel"*, while in 94% of the cases (47 out of 50 users) the "*CL-enhanced, no rel"* algorithm was ranked higher than *"plain CF, no rel"*. In the remaining cases, the algorithm variants not using the confidence level surpassed their counterparts that used the confidence level by a very narrow margin (up to 0.17). Further investigation on the causes that these users exhibited a different stance than the majority of users, including analysis of their profile traits, will be performed in our future work.

Fig. 11 depicts the above-recommender-threshold precision values for the 4 algorithms tested in this experiment. Following the work in (Felfernig, Boratto, Stettinger, & Tkalčič, 2018) and (AlEroud & Karabatis, 2017), we have set the threshold to 7/10 (on our [1-10] rating scale).

We can clearly see that the proposed algorithm, employing the textual-review-to-rating confidence level (computed using the PTD feature) and exploiting the SN user relationships, attains an overall precision of 81.6%, outperforming the "plain CF, with rel" algorithm by 64.5% (49.6% overall precision). Similarly, in SN contexts where no relationships among users are established, the "CL-enhanced, no rel", which uses the textual-review-to-rating confidence level, outperforms the "plain CF, no rel" algorithm (which is its counterpart that does not use the confidence level) by a margin of 23.7% (48% versus 38.8%).

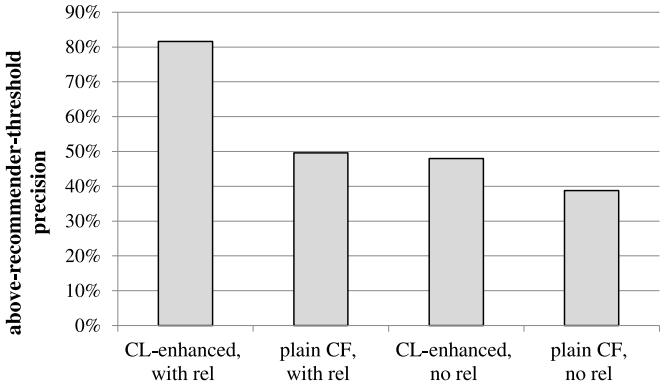


Fig. 11. SN users' recommendation precision

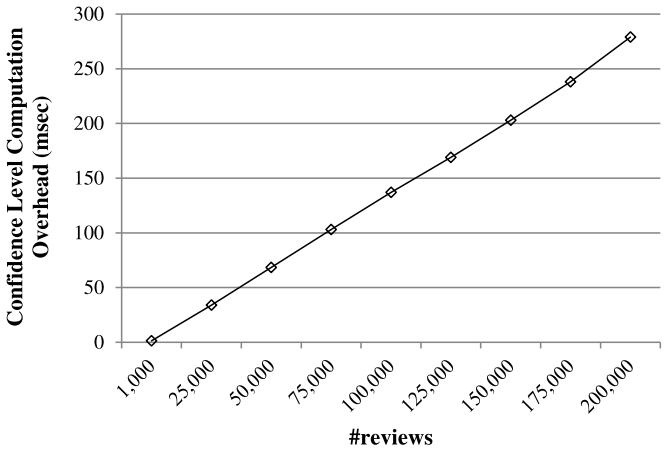


Fig. 12. Confidence level computation overhead for varying numbers of textual reviews

5.4. Performance evaluation

In this subsection we report on the experiments that were designed to calculate the overhead introduced to the prediction process, due to the computation of the confidence level for each individual review-to-rating conversion and the inclusion of this factor to the prediction process.

Fig. 12 depicts the time needed by the proposed algorithm to compute the confidence level for varying numbers of textual reviews, based on their PTD values. The measurements depicted in Fig. 12 correspond to the findings from the Amazon "Videogames" dataset, which has the highest Avg. dataset size / #reviews (= 17), indicating the lengthiest reviews, and consequently the performance overhead introduced by the proposed algorithm is large; hence, the performance metrics in Fig. 12 correspond to a "worst case scenario" (among the tested datasets) for the proposed algorithm.

In Fig. 12 we can clearly see that the aforementioned overhead is small per review (only 0.0014msec) and scales linearly with the number of reviews. Overall, in order to fully process a dataset of the size of the Amazon "Beauty" (200K user reviews), the time required for the used infrastructure is only 280msec, which is deemed fast enough, even for processing in an online fashion. If we execute the aforementioned process on the largest Amazon dataset in existence (which is the Amazon "Books" dataset, containing 8.9M reviews), the overall confidence level computation time has measured to be less than 12sec (recall that there is no need of performing this task online).

6. Conclusions and future work

The work presented in this paper proposes an algorithm that addressed the challenge of computing the reliability level for numeric ratings that have been computed on the basis of textual reviews, through extraction, processing and analyzing textual/linguistic features of the associated review texts. After having examined the usefulness of four different review text features as candidates to be used for improvement of prediction accuracy in the context of CF, it was determined that the polarity term density feature introduced in this work (i.e. the absolute value of the difference between positive and negative terms counts divided by the review length), was the feature most strongly associated with the textual review-to-rating conversion accuracy, surpassing the performance of the document polarity level feature that was proposed in the literature by approximately 3.5 times (44.83% vs. 12.85% rating prediction error reduction), when the Yelp Review Classifier tool was used and by approximately 4 times (31.9% vs. 8.1% rating prediction error reduction) when the VADER Sentiment Analysis tool was used. Subsequently, this metric was used to augment both the computation of user-to-user similarity and the computation of rating predictions, leveraging prediction accuracy, in the CF process. The proposed approach was evaluated in terms of (i) the rating prediction accuracy, (ii) the satisfaction of SN users from the offered recommendations, as well as the recommendation precision, and (iii) the overhead introduced by the computation of the confidence level for each individual review-to-rating conversion, and the results are encouraging. More specifically, as far as rating prediction accuracy is concerned, considering ratios of numeric scores computed from textual reviews to the total number of ratings that range from 10% to 90%, the average improvement is 6.5%; when tested in datasets containing high ratios (90%) of numeric scores computed from textual reviews, the average improvement in rating prediction accuracy was considerably higher (9%). Furthermore, as far as SN users' satisfaction from offered recommendations is concerned, recommendation algorithms that consider the confidence level, proposed in this work, have been shown to increase user satisfaction by a margin ranging from 13.6% to 17.4%, as compared to algorithms that do not assign confidence levels to numeric scores produced from textual rating conversions or use such confidence levels in the rating prediction computation process. The respective improvements, as far as the recommendation precision is concerned, has been computed from 23.7% to 49.6%. Lastly, as far as the computation of the confidence level of the review-to-rating process is concerned (a process which can also be performed offline), this has been quantified to be less than 15msec per 1,000 reviews, on average, for all cases that were examined.

The above measurements clearly indicate that the proposed algorithm achieves considerable rating prediction accuracy gains, while the incurred performance costs is extremely low. Furthermore, the introduction of the confidence level has been shown to deliver performance benefits both in SN where relationships between users can be established, and in SN where such relationships are not present.

Our future work will focus on exploring alternative review characteristics for better capturing the reliability factor in review-to-rating prediction; hence reducing prediction error in CF datasets and improving SN users' recommendations. The performance of the algorithm in datasets where terms expressing polarity may be used to refer to the item content, and not the perceived item quality, will be further analyzed; to this end, more CF datasets will be considered. Additionally, we plan to explore different methods for converting reviews to ratings, including sentiment analysis systems. Finally, the inclusion of valence shifters (words that carry different semantic values than the words described (Balbi, Misuraca, & Scepi, 2018; Vechtomova, 2017)) in our algorithm will be investigated.

CRediT authorship contribution statement

Dionisis Margaris: Conceptualization, Data curation, Methodology, Resources, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Costas Vassilakis:** Conceptualization, Data curation, Methodology, Resources, Validation, Visualization, Writing - original draft, Writing - review & editing. **Dimitris Spiliotopoulos:** Conceptualization, Data curation, Methodology, Resources, Software, Validation, Visualization, Writing - original draft, Writing - review & editing.

References

Abhishek, K. B., Divyashree, M. S., Keerthana, C. S., Meghana, M. K., & Karthik, V. (2019). Review Rating Prediction Using Yelp Dataset. *International Journal of Innovative Science and Research Technology, 4*(1), 712–713.

Ahmadian, S., Meghdadi, M., & Afsharchi, M. (2018). A social recommendation method based on an adaptive neighbor selection mechanism. *Information Processing & Management, 54*(4), 707–725. https://doi.org/10.1016/j.ipm.2017.03.002.

AlEroud, A., & Karabatis, G. (2017). Using Contextual Information to Identify Cyber-Attacks. In I. Alsmadi, G. Karabatis, & A. Aleroud (Vol. Eds.), *Studies in Computational Intelligence: . 691. Information Fusion for Cyber-Security Analytics* (pp. 1–16). Springer. https://doi.org/10.1007/978-3-319-44257-0_1.

Amin, A., Hossain, I., Akther, A., & Alam, K. M. (2019). Bengali VADER: A Sentiment Analysis Approach Using Modified VADER. *Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (pp. 1–6). IEEE. https://doi.org/10.1109/ECACE.2019.8679144.

Bathula, B. (2019). *Predict Yelp ratings. Retrieved from* https://github.com/beegeesquare/Yelp-star-rating.

Bakshy, E., Eckles, D., Yan, R., & Rosenn, I. (2012). Social influence in social advertising: Evidence from field experiments. *Proceedings of the ACM Conference on Electronic Commerce* (pp. 146–161). ACM. https://doi.org/10.1145/2229012.2229027.

Balbi, S., Misuraca, M., & Scepi, G. (2018). Combining different evaluation systems on social media for measuring user satisfaction. *Information Processing & Management, 54*(4), 674–685. https://doi.org/10.1016/j.ipm.2018.04.009.

Bauman, K., Liu, B., & Tuzhilin, A. (2017). Aspect Based Recommendations: Recommending Items with the Most Valuable Aspects Based on User Reviews. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 717–725). ACM. https://doi.org/10.1145/3097983.3098170.

Bellogín, A., & Said, A. (2019). Information Retrieval and Recommender Systems. In A. Said, & V. Torra (Eds.). *Studies in Big Data: 46. Data Science in Practice* (pp. 79–96). Cham.: Springer. https://doi.org/10.1007/978-3-319-97556-6_5.

BrickMover Team. (2019). *RecSys BrickMover's code.* Retrieved from https://kaggle2.blob.core.windows.net/forum-message-attachments/9420/RecSys_BrickMovers_Source_Code.pdf.

Camacho, L. A. G., & Alves-Souza, S. N. (2018). Social network data to alleviate cold-start in recommender system: A systematic review. *Information Processing & Management, 54*(4), 529–544. https://doi.org/10.1016/j.ipm.2018.03.004.

Chen, J., Uliji, Wang, H., & Yan, Z. (2018). Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering. *Swarm and Evolutionary Computation, 38*, 35–41. https://doi.org/10.1016/j.swevo.2017.05.008.

Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction, 25*(2), 99–154. https://doi.org/10.1007/s11257-015-9155-5.

Chikersal, P., Poria, S., Cambria, E., Gelbukh, A., & Siong, C. E. (2015). Modelling Public Sentiment in Twitter: Using Linguistic Patterns to Enhance Supervised Learning. In A. Gelbukh (Ed.). *Computational Linguistics and Intelligent Text Processing* (pp. 49–65). Springer International Publishing.

Cieslik, J. (2019). *nyyelp: predicting yelp review rating using recurrent neural networks. Retrieved from.* https://github.com/i008/nyyelp.

Desrosiers, C., & Karypis, G. (2011). A Comprehensive Survey of Neighborhood-based Recommendation Methods. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.). *Recommender Systems Handbook* (pp. 107–144). Springer US. https://doi.org/10.1007/978-0-387-85820-3_4.

Felfernig, A., Boratto, L., Stettinger, M., & Tkalčič, M. (2018). Evaluating Group Recommender Systems. *In Group Recommender Systems. SpringerBriefs in Electrical and Computer Engineering,* 59–71. https://doi.org/10.1007/978-3-319-75067-5_3.

Frémal, S., & Lecron, F. (2017). Weighting strategies for a recommender system using item clustering based on genres. *Expert Systems with Applications, 77*, 105–113. https://doi.org/10.1016/j.eswa.2017.01.031.

Ganesan, K., & Zhai, C. (2012). Opinion-based entity ranking. *Information Retrieval, 15*(2), 116–150. https://doi.org/10.1007/s10791-011-9174-8.

Ganu, G., Kakodkar, Y., & Marian, A. (2013). Improving the quality of predictions using textual information in online user reviews. *Information Systems, 38*(1), 1–15. https://doi.org/10.1016/j.is.2012.03.001.

Hassani, A., Haghighi, P. D., Ling, S., Jayaraman, P. P., & Zaslavsky, A. (2018). Querying IoT services: A smart carpark recommender use case. *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)* (pp. 619–624). IEEE. https://doi.org/10.1109/WF-IoT.2018.8355158.

He, R., Kang, W.-C., & McAuley, J. (2017). Translation-Based Recommendation. *Proceedings of the Eleventh ACM Conference on Recommender Systems* (pp. 161–169). ACM. https://doi.org/10.1145/3109859.3109882.

Hu, Y.-H., Chen, Y.-L., & Chou, H.-L. (2017). Opinion mining from online hotel reviews – A text summarization approach. *Information Processing & Management, 53*(2), 436–449. https://doi.org/10.1016/j.ipm.2016.12.002.

Hutto, C. (2019). *VADER Sentiment Analysis. Retrieved from.* https://github.com/cjhutto/vaderSentiment.

Jadidinejad, A. H., Macdonald, C., & Ounis, I. (2019). Unifying Explicit and Implicit Feedback for Rating Prediction and Ranking Recommendation Tasks. *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval - ICTIR '19* (pp. 149–156). ACM. https://doi.org/10.1145/3341981.3344225.

Karimi, M., Jannach, D., & Jugovac, M. (2018). News recommender systems – Survey and roads ahead. *Information Processing & Management, 54*(6), 1203–1227. https://doi.org/10.1016/j.ipm.2018.04.00810.1016/j.ipm.2018.04.008.

Koren, Y., & Bell, R. (2011). Advances in Collaborative Filtering. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.). *Recommender Systems Handbook* (pp. 145–186). Springer US. https://doi.org/10.1007/978-0-387-85820-3_5.

Kronmueller, M., Chang, D., Hu, H., & Desoky, A. (2018). A Graph Database of Yelp Dataset Challenge 2018 and Using Cypher for Basic Statistics and Graph Pattern Exploration. *Proceedings of the 18th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT 2018)* (pp. 135–140). IEEE. https://doi.

org/10.1109/ISSPIT.2018.8642700.
Lee, D. H., & Brusilovsky, P. (2017). Improving personalized recommendations using community membership information. *Information Processing & Management, 53*(5), 1201–1214. https://doi.org/10.1016/j.ipm.2017.05.005.
Lester, D. (2019). *Yelp Rating and Review Trends. Retrieved from.* https://github.com/davelester/Yelp-Rating-and-Review-Trends.
Li, C.-T., Shan, M.-K., Jheng, S.-H., & Chou, K.-C. (2016). Exploiting concept drift to predict popularity of social multimedia in microblogs. *Information Sciences, 339*, 310–331. https://doi.org/10.1016/j.ins.2016.01.009.
Liu, B. (2019). *Opinion lexicon (or sentiment lexicon)*. https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html Accessed 24 Nov 2019.
Logesh, R., Subramaniyaswamy, V., Vijayakumar, V., & Li, X. (2019). Efficient User Profiling Based Intelligent Travel Recommender System for Individual and Group of Users. *Mobile Networks and Applications, 24*, 1018–1033. https://doi.org/10.1007/s11036-018-1059-2.
Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.). *Recommender Systems Handbook* (pp. 73–105). Springer US. https://doi.org/10.1007/978-0-387-85820-3_3.
Margaris, D., & Vassilakis, C. (2017a). Improving Collaborative Filtering's Rating Prediction Quality by Considering Shifts in Rating Practices. *Proceedings of the 19th IEEE Conference on Business Informatics (CBI 2017), 1*, 158–166. https://doi.org/10.1109/CBI.2017.24.
Margaris, D., & Vassilakis, C. (2016). Pruning and aging for user histories in collaborative filtering. *2016 IEEE Symposium Series on Computational Intelligence (SSCI),* 1–8. https://doi.org/10.1109/SSCI.2016.7849920.
Margaris, D., & Vassilakis, C. (2017). Exploiting Internet of Things information to enhance venues' recommendation accuracy. *Service Oriented Computing and Applications, 11*(4), 393–409. https://doi.org/10.1007/s11761-017-0216-y.
Margaris, D., & Vassilakis, C. (2017). Enhancing User Rating Database Consistency Through Pruning. *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXIV, Lecture Notes in Computer Science, 10620*, 33–64. https://doi.org/10.1007/978-3-662-55947-5_3.
Margaris, D., & Vassilakis, C. (2018). Exploiting Rating Abstention Intervals for Addressing Concept Drift in Social Network Recommender Systems. *Informatics, 5*(2), 21. https://doi.org/10.3390/informatics5020021.
Margaris, D., Georgiadis, P., & Vassilakis, C. (2015). A collaborative filtering algorithm with clustering for personalized web service selection in business processes. *Proceedings of the 9th IEEE International Conference on Research Challenges in Information Science (RCIS 2015)* (pp. 169–180). IEEE. https://doi.org/10.1109/RCIS.2015.7128877.
Margaris, D., & Vassilakis, C. (2018). Improving collaborative filtering's rating prediction accuracy by considering users' rating variability. *Proceedings of the 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress* (pp. 1022–1027). IEEE. https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00145.
Margaris, D., Vassilakis, C., & Georgiadis, P. (2017). *Knowledge-Based Leisure Time Recommendations in Social Networks. Current Trends on Knowledge-Based Systems*. Springer23–48. https://doi.org/10.1007/978-3-319-51905-0_2.
Margaris, D., Vassilakis, C., & Georgiadis, P. (2018). Query personalization using social network information and collaborative filtering techniques. *Future Generation Computer Systems, 78*, 440–450. https://doi.org/10.1016/j.future.2017.03.015.
Margaris, D., Vassilakis, C., & Spiliotopoulos, D. (2019). Handling uncertainty in social media textual information for improving venue recommendation formulation quality in social networks. *Social Network Analysis and Mining, 9*(1), 64. https://doi.org/10.1007/s13278-019-0610-x.
Margaris, D., Vassilakis, C., & Spiliotopoulos, D. (2020). *Making recommendations in Social Networks based on textual reviews: a confidence-based approach (version 2.0)*. Software and Database Systems Laboratory, University of the Peloponnesehttps://soda.dit.uop.gr/?q=TR-19002-v2.
Marinho, L. B., Nanopoulos, A., Schmidt-Thieme, L., Jäschke, R., Hotho, A., Stumme, G., & Symeonidis, P. (2011). Social Tagging Recommender Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.). *Recommender Systems Handbook* (pp. 615–644). Springer US. https://doi.org/10.1007/978-0-387-85820-3_19.
McAuley, J., Pandey, R., & Leskovec, J. (2015). Inferring Networks of Substitutable and Complementary Products. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15* (pp. 785–794). ACM. https://doi.org/10.1145/2783258.2783381.
McAuley, J., Targett, C., Shi, Q., & van den Hengel, A. (2015). Image-Based Recommendations on Styles and Substitutes. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15* (pp. 43–52). ACM. https://doi.org/10.1145/2766462.2767755.
Musto, C., de Gemmis, M., Semeraro, G., & Lops, P. (2017). A Multi-criteria Recommender System Exploiting Aspect-based Sentiment Analysis of Users' Reviews. *Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17* (pp. 321–325). ACM. https://doi.org/10.1145/3109859.3109905.
Najafabadi, M. K., Mohamed, A., & Onn, C. W. (2019). An impact of time and item influencer in collaborative filtering recommendations using graph-based model. *Information Processing & Management, 56*(3), 526–540. https://doi.org/10.1016/j.ipm.2018.12.007.
Ngo-Ye, T. L., Sinha, A. P., & Sen, A. (2017). Predicting the helpfulness of online reviews using a scripts-enriched text regression model. *Expert Systems with Applications, 71*, 98–110. https://doi.org/10.1016/j.eswa.2016.11.029.
Nilashi, M., Ibrahim, O., Yadegaridehkordi, E., Samad, S., Akbari, E., & Alizadeh, A. (2018). Travelers decision making using online review in social network sites: A case on TripAdvisor. *Journal of Computational Science, 28*, 168–179. https://doi.org/10.1016/j.jocs.2018.09.006.
Obar, J. A., & Wildman, S. S. (2015). Social Media Definition and the Governance Challenge: An Introduction to the Special Issue. *SSRN Electronic Journal.* https://doi.org/10.2139/ssrn.2647377.
Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP '02. 10. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP '02* (pp. 79–86). Association for Computational Linguistics. https://doi.org/10.3115/1118693.1118704.
Paradarami, T. K., Bastian, N. D., & Wightman, J. L. (2017). A hybrid recommender system using artificial neural networks. *Expert Systems with Applications, 83*, 300–313. https://doi.org/10.1016/j.eswa.2017.04.046.
Pasricha, R., & McAuley, J. (2018). Translation-Based Factorization Machines for Sequential Recommendation. *Proceedings of the 12th ACM Conference on Recommender Systems* (pp. 63–71). ACM. https://doi.org/10.1145/3240323.3240356.
Pereira, R. B., Plastino, A., Zadrozny, B., & Merschmann, L. H. C. (2018). Correlation analysis of performance measures for multi-label classification. *Information Processing & Management, 54*(3), 359–369. https://doi.org/10.1016/j.ipm.2018.01.002.
Qian, Y., Zhang, Y., Ma, X., Yu, H., & Peng, L. (2019). EARS: Emotion-aware recommender system based on hybrid information fusion. *Information Fusion, 46*, 141–146. https://doi.org/10.1016/j.inffus.2018.06.004.
Qiu, J., Liu, C., Li, Y., & Lin, Z. (2018). Leveraging sentiment analysis at the aspects level to predict ratings of reviews. *Information Sciences,* 451–452. https://doi.org/10.1016/j.ins.2018.04.009 295–309.
Sánchez, P., & Bellogín, A. (2019). Building user profiles based on sequences for content and collaborative filtering. *Information Processing & Management, 56*(1), 192–211. https://doi.org/10.1016/j.ipm.2018.10.003.
Seo, S., Huang, J., Yang, H., & Liu, Y. (2017). Representation Learning of Users and Items for Review Rating Prediction Using Attention-based Convolutional Neural Network. *Proceedings of the 3rd International Workshop on Machine Learning Methods for Recommender Systems* (pp. 1–8). .
Tran, K. (2019). Yelp Review Classifier. Retrieved from https://github.com/kevintrankt/yelp-review-classifier.
Ureña, R., Kou, G., Dong, Y., Chiclana, F., & Herrera-Viedma, E. (2019). A review on trust propagation and opinion dynamics in social networks and group decision making frameworks. *Information Sciences, 478*, 461–475. https://doi.org/10.1016/j.ins.2018.11.037.
Valcarce, D., Bellogín, A., Parapar, J., & Castells, P. (2018). On the robustness and discriminative power of information retrieval metrics for top-N recommendation. *Proceedings of the 12th ACM Conference on Recommender Systems - RecSys '18* (pp. 260–268). ACM. https://doi.org/10.1145/3240323.3240347.
Valverde-Rebaza, J. C., Roche, M., Poncelet, P., & de Andrade Lopes, A. (2018). The role of location and social strength for friendship prediction in location-based social networks. *Information Processing & Management, 54*(4), 475–489. https://doi.org/10.1016/j.ipm.2018.02.004.
Vechtomova, O. (2017). Disambiguating context-dependent polarity of words: An information retrieval approach. *Information Processing & Management, 53*(5), 1062–1079. https://doi.org/10.1016/j.ipm.2017.03.007.
Vijayakumar, V., Vairavasundaram, S., Logesh, R., & Sivapathi, A. (2019). Effective Knowledge Based Recommender System for Tailored Multiple Point of Interest Recommendation. *International Journal of Web Portals, 11*(1), 1–18. https://doi.org/10.4018/IJWP.2019010101.

Wang, Y., Liu, Y., & Yu, X. (2012). Collaborative Filtering with Aspect-Based Opinion Mining: A Tensor Factorization Approach. *Proceedings of the IEEE 12th International Conference on Data Mining (ICDM 2012),* 1152–1157. https://doi.org/10.1109/ICDM.2012.76.

Xing, F. Z., Pallucchini, F., & Cambria, E. (2019). Cognitive-inspired domain adaptation of sentiment lexicons. *Information Processing & Management, 56*(3), 554–564. https://doi.org/10.1016/j.ipm.2018.11.002.

Xu, C. (2018). A novel recommendation method based on social network using matrix factorization technique. *Information Processing & Management, 54*(3), 463–474. https://doi.org/10.1016/j.ipm.2018.02.005.

Yi, J., Huang, J., & Qin, J. (2018). Rating Prediction in Review-Based Recommendations via Adversarial Auto-Encoder. *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI),* 144–149. https://doi.org/10.1109/WI.2018.00-96.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph Convolutional Neural Networks for Web-Scale Recommender Systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18* (pp. 974–983). ACM. https://doi.org/10.1145/3219819.3219890.

Zhang, S., Zhang, X., Chan, J., & Rosso, P. (2019). Irony detection via sentiment-based transfer learning. *Information Processing & Management, 56*(5), 1633–1644. https://doi.org/10.1016/j.ipm.2019.04.006.

Zheng, L., Noroozi, V., & Yu, P. S. (2017). Joint Deep Modeling of Users and Items Using Reviews for Recommendation. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining - WSDM '17* (pp. 425–434). ACM. https://doi.org/10.1145/3018661.3018665.

Zhou, W., & Han, W. (2019). Personalized recommendation via user preference matching. *Information Processing & Management, 56*(3), 955–968. https://doi.org/10.1016/j.ipm.2019.02.002.