A Context-Aware Recommender Method Based on Text Mining

Camila Vaccari Sundermann[1](✉), Renan de Pádua[1], Vítor Rodrigues Tonon[1],
Marcos Aurélio Domingues[2], and Solange Oliveira Rezende[1]

[1] Institute of Mathematics and Computer Science - University of São Paulo,
Avenida Trabalhador São Carlense, 400, São Carlos, SP 13566-590, Brazil
{camilavs,padua,vitor.tonon,solange}@icmc.usp.br

[2] Department of Informatics, State University of Maringá,
Avenida Colombo, 5790, Maringá, PR 87020-900, Brazil
madomingues@uem.br

Abstract. A recommender system is an information filtering technology that can be used to recommend items that may be of interest to users. In their traditional form, recommender systems do not consider information that might enrich the recommendation process, as contextual information. In this way, we have the context-aware recommender systems that consider contextual information to generate the recommendations. Reviews can provide relevant information that can be used by recommender systems, including the contextual one. Thus, in this paper, we propose a context-aware recommender method based on text mining (**CARM-TM**) that includes two context extraction techniques: (1) $CIET.5_{embed}$, a technique based on word embeddings; and (2) *RulesContext*, a technique based on association rules. For this work, **CARM-TM** makes use of context by running the CAMF algorithm, a context-aware recommender system based on matrix factorization. To evaluate our method, we compare it against the MF algorithm, an uncontextual recommender system based on matrix factorization. The evaluation showed that our method presented better results than the MF algorithm in most cases.

Keywords: Context-aware recommender systems · Text mining · Association rules · Word embedding · Matrix factorization

1 Introduction

A recommender system is an information filtering technology that can be used to predict ratings for items (like products, services, *etc.*) and/or generate a custom item ranking that may be of interest to the user [1]. In their traditional form, recommender systems consider only the items that users have accessed, bought or evaluated positively, thus ignoring any other information that might enrich the recommendation process. One type of information that may enrich the process

P. Moura Oliveira et al. (Eds.): EPIA 2019, LNAI 11805, pp. 385–396, 2019.
https://doi.org/10.1007/978-3-030-30244-3_32

is contextual information. For example, when recommending a restaurant to a user, the system may consider the context "Day of the Week". At weekends the user may prefer snack bars while on other days he/she may prefer less caloric meals.

In this way, there are the context-aware recommender systems that, unlike traditional systems, also consider contextual information to generate the set of recommendations. The term "context" may assume different definitions depending on the area in which it is being used. In the area of recommender systems, the definition most used and adopted in our work was proposed by Dey [8]. According to this author, "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

Many authors have been working with new context-aware recommender algorithms. However, there is a lack of automatic techniques for extracting context. With the emergence of Web 2.0, users have enriched sites with contextual information through texts in social networks, comments and mainly through reviews. These reviews are usually in the form of textual comments, in which users explain why they liked or disliked an item based on their own experiences. According to Chen *et al.* [7], the incorporation of important information extracted from reviews can benefit the recommender systems, solving the sparse data and the cold-start problems. Reviews can provide relevant information that can be used by recommender systems, including the contextual one.

Thus, in this work, we propose a context-aware recommender method based on text mining (**CARM-TM**) that includes two context extraction techniques: (1) $CIET.5_{embed}$, a technique based on word embeddings; and (2) *RulesContext*, a technique based on association rules. **CARM-TM** makes use of context by running the CAMF algorithm, a context-aware recommender based on matrix factorization. To evaluate our method, we compare it against the MF algorithm, an uncontextual recommender system based on matrix factorization. The evaluation was conducted in a dataset of reviews from Yelp and showed that our method provided better results than the MF algorithm in most cases.

This paper is structured as follows: in Sect. 2, we describe some related works about context extraction based on text mining. In Sect. 3, we present our proposal, a context-aware recommender method based on text mining (**CARM-TM**). We evaluate and discuss the main findings in Sect. 4. Finally, in Sect. 5, we present conclusion and future work.

2 Related Work

In this section, we present some related works that extract context from reviews or from other textual sources.

Li et al. [14] investigated available restaurant reviews and four types of contextual information for a meal. They developed algorithms with existing natural language processing tools to extract these types of contextual information from restaurant reviews. Hariri et al. [9] obtained contextual information by mining hotel reviews written by users. Their approach is based on using a classifier which is trained by the description sample and their corresponding contexts.

In [3], Bauman and Tuzhilin presented a method to find relevant contextual information from reviews of users. In this method, the reviews are classified as "specifics" and "generics", and the context is extracted from the specific reviews by using two methods: "word-based" and "LDA-based". Chen and Chen [6] extracted contexts employing a keyword matching method.

Kim et al. [10] presented a recommendation system model called Convolutional Matrix Factorization (ConvMF). The model integrates convolutional neural networks into probabilistic matrix factorization in order to capture contextual information (adjacent words) of the documents. Sulthana and Ramasamy [15] proposed an Ontology and Context Based Recommendation System for the book domain that uses a Neuro-Fuzzy Classification approach.

In [16], we proposed the $CIET.5_{embed}$, a textual context extraction technique based on word embeddings model that was used with neighborhood-based contextual recommender systems. This technique is implemented in our **CARM-TM**, so it is detailed in Sect. 3.4. In addition, in this paper, we also propose for the **CARM-TM**, the *RulesContext* technique, that extracts association rules from user reviews and transforms them into contextual information to be used in recommender systems. In the next section, we present our context-aware recommender method (**CARM-TM**).

3 Context-Aware Recommender Method Based on Text Mining (CARM-TM)

In this work, we propose the **CARM-TM**, a context-aware recommender method that uses text mining techniques to extract contextual information from reviews to make recommendations. The CARM-TM, illustrated in Fig. 1, has 5 steps which are explained in the next subsections.

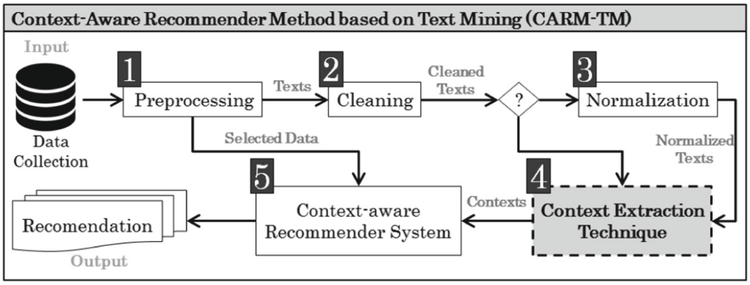

Fig. 1. Overview of the Context-Aware Recommender Method based on Text Mining (**CARM-TM**).

3.1 Step 1 - Preprocessing

The input of our method is a dataset, where which row contains a user identification, the identification of the item evaluated by the user together with the evaluation value, a textual content containing reviews/opinions about the item, and the date when the evaluation was made.

The step 1 is responsible for preparing the dataset for both the context extraction and the recommendation steps. In this step, the data are filtered, excluding those without textual content or other important information such as the user or item identification. In addition, users, items and reviews that are less relevant are excluded by using the exclusion criterias in [6]: (1) users with 1 review; (2) items with less than 15 reviews; and (3) reviews with less than 3 sentences. Besides filtering, we also create a file for each review.

3.2 Step 2 - Cleaning

In the step 2, the textual content goes through a cleaning in order to eliminate special characters such as @, $*$, # and &. These characters may negatively influence the context extraction process. Then, the cleaned texts can pass through a normalizer in the step 3 or they can be directly used by the context extraction technique (step 4).

3.3 Step 3 - Normalization

Normalization is optional and aims to solve problems commonly encountered in texts written by users, like typos, spelling mistakes, abbreviations, etc. In this work, we used the TextExpansion[1] tool to normalize the texts.

3.4 Step 4 - Context Extraction

The main step of our method is the fourth step, which consists of extracting contextual information from reviews. Here, we can adopt different text mining techniques. For this work, we use the $CIET.5_{embed}$ technique, proposed by us in [16]. In addition, we also propose the *RulesContext*, a new technique for context extraction that extracts association rules from reviews to be used as contextual information in recommender systems. Both techniques are detailed in the following subsections.

Contextual Information Extraction Technique Based on Word Embeddings ($CIET.5_{embed}$). Proposed by us in [16], this technique consists of combining two types of representations (bag of words and word embedding model) that allow to raise the volume and quality of information, the latent relationships among terms from documents, and the interpretability of the generated text representations. The $CIET.5_{embed}$ technique is composed of five complementary steps (Fig. 2), which aim to transform a set of text documents into a set of contextualized documents.

[1] http://lasid.sor.ufscar.br/expansion/static/index.html.

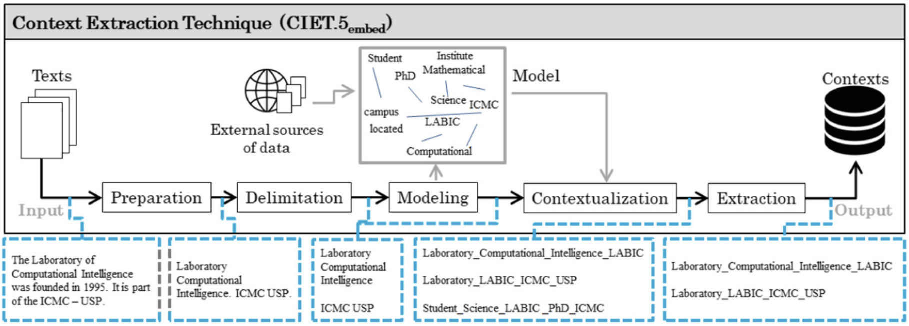
Context Extraction Technique (CIET.5embed)
Texts
External sources of data
Student
Institute
PhD
Mathematical
Science
ICMC
campus
located
LABIC
Computational
Model
Contexts
Input
Preparation
Delimitation
Modeling
Contextualization
Extraction
Output
The Laboratory of Computational Intelligence was founded in 1995. It is part of the ICMC – USP.
Laboratory Computational Intelligence. ICMC USP.
Laboratory Computational Intelligence
ICMC USP
Laboratory_Computational_Intelligence_LABIC
Laboratory_LABIC_ICMC_USP
Student_Science_LABIC_PhD_ICMC
Laboratory_Computational_Intelligence_LABIC
Laboratory_LABIC_ICMC_USP

Fig. 2. Overview of the Context Extraction Technique $CIET.5_{embed}$. Adapted from Sundermann et al. [16].

In Fig. 2, the **Preparation** step sends all documents to a textual enrichment, which consists of named entity and concept recognition. In the **Delimitation** step, the documents already prepared are submitted to a process to delimit the textual scopes, like paragraphs and sentences. In the **Modeling**, a language model based on word embeddings, previously trained with external source of documents, is retrained with the internal documents. Then, we identify contexts in the internal documents (**Contextualization** step). The terms of each sentence are processed in the language model to find their most related terms, by using for example the cosine measure. Finally, in the step **Extraction**, the contexts are extracted from the documents by using a comparative threshold.

Contextual Information Extraction Technique Based on Association Rules (*RulesContext*). Proposed in this work, the *RulesContext* is a technique that extracts association rules from reviews and transform them into contextual information to be used in context-aware recommender systems.

Association rules are widely used in the literature to find correlation among items on a given database [2]. The association rules are presented on the format $LHS \rightarrow RHS$, where LHS stands for left hand side and RHS for right hand side, both of them contains a set of items such as $LHS \cap RHS = \emptyset$.

The *RulesContext* technique is executed in four steps, as illustrated in Fig. 3. In the step **Separation by item**, the texts are separated by item, i.e. subsets of texts are grouped for each item that can be recommended. Each subset is composed of the reviews' texts about the item.

In the second step (**Preprocessing and Preparation**), the texts are preprocessed, i.e. the stopwords are removed and the terms are stemmed. Besides, each subset of texts is transformed into a transaction.

In the third step (**Extraction of Association Rules**), we extract the association rules from each subset. To extract the rules, we use the algorithm apriori [2]. This algorithm extracts the rules in 2 steps by combining the items on a given dataset and calculating the measures support and confidence for each rule. After extracting the rules, we use the mutual information (MI) measure to evaluate

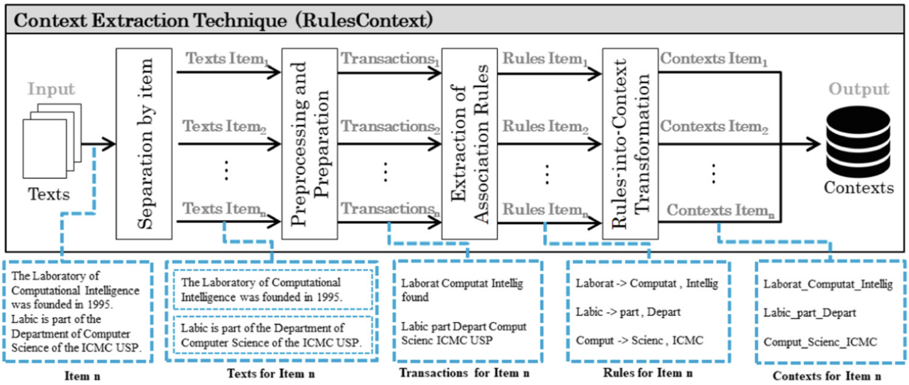
Context Extraction Technique (RulesContext)
Input
Texts
Separation by item
Texts Item$_1$
Texts Item$_2$
Texts Item$_n$
Preprocessing and Preparation
Transactions$_1$
Transactions$_2$
Transactions$_n$
Extraction of Association Rules
Rules Item$_1$
Rules Item$_2$
Rules Item$_n$
Rules-into-Context Transformation
Contexts Item$_1$
Contexts Item$_2$
Contexts Item$_n$
Output
Contexts
The Laboratory of Computational Intelligence was founded in 1995. Labic is part of the Department of Computer Science of the ICMC USP.
Item n
The Laboratory of Computational Intelligence was founded in 1995.
Labic is part of the Department of Computer Science of the ICMC USP.
Texts for Item n
Laborat Computat Intellig found
Labic part Depart Comput Scienc ICMC USP
Transactions for Item n
Laborat -> Computat , Intellig
Labic -> part , Depart
Comput -> Scienc , ICMC
Rules for Item n
Laborat_Computat_Intellig
Labic_part_Depart
Comput_Scienc_ICMC
Contexts for Item n

Fig. 3. Overview of the *RulesContext* technique.

them. The MI measures how dependent the items on the LHS and RHS are. The mutual information is presented in Eq. 1.

$$MI(LHS \rightarrow RHS) = Support(LHS \cup RHS) log(Lift(LHS \rightarrow RHS)). \quad (1)$$

After extracting the association rules, in the step four **Rules-into-Context Transformation**, the rules are transformed into contextual information as illustrated in Fig. 3.

3.5 Step 5 - Context-Aware Recommender Systems

In the fifth step of the **CARM-TM**, the contextual dataset generated by one of the previous Context Extraction Techniques is used as input by a **Context-Aware Recommender System**, together with the user and data items obtained in the Preprocessing. Contextual information is considered for recommendation according to the type of recommender system that is being used by the method. Latent factor models look for finding hidden features or patterns in the training data, also called factors, that are used to make the recommendations. Some of the most successful latent factor models are based upon matrix factorization techniques, such as the one presented by [11], which combines good accuracy and scalability. For this reason, in this paper, we use, as baseline, the matrix factorization algorithm (MF) [12], and as context-aware system, the context-aware matrix factorization algorithm (CAMF) [1].

According to Aggarwal [1], the recommendation training data consists of ratings given by users to sets of items, which are organized into the ratings matrix R. This matrix, given m users and n items, is of size $m \times n$ and the entry r_{x_i,y_j} corresponds to the rating given by user x_i to item y_j. The main purpose of matrix factorization is to decompose this matrix, R, into two approximate smaller matrices, X and Y, seeking to find k latent factors, which are hidden features or patterns in the training data, for the m users and the n items, respectively.

Considering that our purpose is to predict the unknown ratings in the matrix R, it is possible to use the inferred matrices X and Y to compute an approximate rating prediction. The predicted rating $\hat{r}_{x_i,y_j}$ is given by the cross-product of the user-factors vector and the item-factors vector, as shown in Eq. 2, such that $\boldsymbol{x}_i$ corresponds to the factors inferred for user x_i and $\boldsymbol{y}_j$ to the factors inferred for item y_j.

$$\hat{r}_{x_i,y_j} = \boldsymbol{x}_i \cdot (\boldsymbol{y}_j)^T \tag{2}$$

In order to obtain the factors vectors, the system should minimize Eq. 3, by using the training data (set S) and some optimization algorithm, such as *Stochastic Gradient Descent* [4] or *Alternating Least Squares* [13]. Regardless of the chosen algorithm, the parameters k and λ must be optimized. The first parameter corresponds to the number of latent features used to model the recommendation data. It is responsible for making the model simpler or more complex, depending on how much complexity is needed to capture all of the latent dimensions of the input data. The second parameter (λ) is used to weight the regularization constraint, in order to prevent overfitting. This algorithm is usually called MF, i.e. matrix factorization algorithm.

$$J = \sum_{(x_i,y_j)\in S} (r_{x_i,y_j} - \boldsymbol{x}_i \cdot (\boldsymbol{y}_j)^T)^2 + \lambda(\|\boldsymbol{x}_i\|^2 + \|\boldsymbol{y}_j\|^2) \tag{3}$$

Matrix factorization techniques are not exclusive to traditional recommender systems, Aggarwal [1] describes a method based on pairwise interactions that is suited to the context-aware recommendation task. The central idea in pairwise interaction algorithms is to decompose the ratings tensor R into n factor matrices, such that the first two correspond to users (U) and items (V) and the others correspond to the contextual variables (C_a, $1 \leq a \leq n-2$). This new matrices are then used to make the rating prediction ($\hat{r}_{i,j,c_1\cdots,c_{n-2}}$) for user i, item j and contexts $c_1, \cdots c_{n-2}$, by multiplying them in a pairwise manner, as shown in Eq. 4.

$$\hat{r}_{i,j,c_1\cdots,c_{n-2}} = (UV^T)_{ij} + (UC_1^T)_{ic_1} + (UC_2^T)_{ic_2} + \cdots + (C_{n-3}C_{n-2}^T)_{c_{n-3}c_{n-2}} \tag{4}$$

In order to obtain this matrices, the following equation must be minimized (Eq. 5) using some optimization algorithm. The parameter λ is used for regularization purposes and the set S consists of specified ratings. This algorithm is called CAMF, i.e. context-aware matrix factorization algorithm

$$J = \sum_{(i,j,c_1\cdots,i_{n-2})\in S} (r_{i,j,c_1\cdots,c_{n-2}} - \hat{r}_{i,j,c_1\cdots,c_{n-2}})^2 + \lambda(||U||^2 + ||V||^2 + \sum_{a=1}^{n-2} ||C_a||^2) \tag{5}$$

The output of our context-aware recommender method is the recommendations generated by the context-aware recommender systems using the two types of context extracted by the *CIET.5*$_{embed}$ and the *RulesContext* techniques. In the next section we present the empirical evaluation conducted with our proposal.

4 Empirical Evaluation

For this work, we carried out two different evaluations. In the first one, we compare the CARM-TM method, with the *CIET.5*$_{embed}$ and the *RulesContext* techniques, against the uncontextual MF algorithm (baseline). With this evaluation, we aimed to demonstrate the impact of the use of contextual information extracted by the *CIET.5*$_{embed}$ and *RulesContext* techniques in the contextual CAMF recommender systems. Additionally, in the second evaluation, we use the CARM-TM method to compare the *CIET.5*$_{embed}$ against the *RulesContext*, in order to identify which contextual extraction technique provides the best recommendations.

4.1 Dataset

The dataset used in the empirical evaluation was the RecSys dataset for the recommender system challenge, ACM RecSysChallenge 2013, proposed to the customization of recommendations for Yelp[2] users. In the Yelp website the users can evaluate businesses through reviews. In these reviews, it is possible to evaluate the item by leaving a rating in the format of stars (from one to five stars). In addition, the user can write a text explaining his/her opinion about the establishment and the reason for which he/she gave a certain note. The RecSys dataset contains 11,537 items (businesses), 45,980 users and 229,901 reviews.

4.2 Experimental Setup

To measure the predictive ability of the recommender systems, we used the *All But One protocol* [5] with 10-fold cross validation, where the set of documents were partitioned into 10 subsets. For each fold, we used $n - 1$ of these subsets for training and the rest for testing. The training set T_r was used to build the recommendation model. For each user in the test set T_e, an item was hidden as a singleton set H, and the remaining items represent the set of observable items O used in the recommendation. Based on 10-fold cross validation, we computed *Mean Average Precision* for 10 recommendations (MAP@10) and to compare two recommendation algorithms, we applied the two-sided paired t-test with a 95% confidence level.

For the *CIET.5*$_{embed}$ technique, we considered the threshold values: 0.25, 0.50 and 1.0. The context sizes were 4 and 10 words. Altogether, we used 6 different configurations (3 threshold values × 2 context sizes). These values were adopted according to the best results obtained in our previous work [16].

Regarding the *RulesContext* technique, we generated the association rules using a minimum support value equals to 10% and a confidence value equals to 50%. To select the most relevant rules, we used cut percentages of the MI measure equal to 50% and 75%. From the most relevant rules, we selected sets

[2] https://www.yelp.com.

with the top 5, 10 and 20 rules, totalling 6 combinations (2 MI values × 3 set sizes).

The MF and CAMF algorithms were executed 20 times, varying the values of k and λ (see Sect. 3.5). The used values were: $k = 5$, 10, 50 and 100; and $\lambda = 1$, 10, 100, 150 and 200.

4.3 Results

In this section, we first present the results of our proposal with the $CIET.5_{embed}$, and then, with the *RulesContext* technique. Both against the baseline MF. In Tables 1 and 2, the values that are statistically significant (p-value > 0.05) are with an asterisk and the highest values are in boldface. At the end, we compare the results between the $CIET.5_{embed}$ and the *RulesContext* techniques.

Table 1 presents the results of our proposal, with the $\boldsymbol{CIET.5_{embed}}$ **technique** and the CAMF algorithm, against the baseline MF. We refer to each contextual information as *Size_Threshold*. For example, the type of context 4_025 represents the context with 4 words that were extracted considering the value of threshold 0.25. In Table 1, we can observe that, in general, the results were very satisfactory compared against the baseline MF. Contextual information 10_1 provided the best results. However, the context *10_025* presented the best value of MAP@10 for the parameters k equals to 5 and λ equals to 150. We must emphasize that this combination of parameters resulted the best results for these experiments in particular.

The results of our proposal, with the contexts extracted by the ***RulesContext technique***, are presented in Table 2. There, we refer to each contextual information as *MI_NumberOfRules*. For example, the type of context *50_5* means the top 5 rules (contexts) extracted using the cut percentage of the MI measure equals to 50%. Again, the results were very satisfactory. We can observe that the contexts extracted with the cut of MI equals to 50% presented the best results in most of the cases, with the highest value of MAP being provided by the context *50_20*.

Analyzing the parameters used in the experiments, we observed that for $CIET.5_{embed}$, the best result was obtained with a context size equals to 10 and a threshold equals to 25%. For the *RulesContext* technique, the MI cut that generated the best results was 50%, that is, using a greater number of rules and selecting the best ones. Regarding the matrix factorization parameters, the best results were obtained with k equals to 5 and λ equals to 150, that is, a model with low complexity level and with a relatively high value of the parameter that controls overfitting.

Finally, we present in Fig. 4 the comparison between the $CIET.5_{embed}$ and the *RulesContext* techniques. There, we compare the best MAP@10 (vertical axis) varying the values of λ and k (horizontal axis). In most cases, the $CIET.5_{embed}$ performed better than the *RulesContext* technique. However, in two cases, *10_5* e *150_5*, the *RulesContext* technique was superior. In Fig. 4, we can also observe that our proposal with the context extraction techniques outperformed the baseline MF in all cases.

Table 1. Comparing the results ($MAP@10$) of our proposal, with the CAMF contextual recommender algorithm using contexts extracted by the $CIET.5_{embed}$ technique, against the results of the baseline MF.

Parameters		Baseline	4_025	4_05	4_1	10_025	10_05	10_1
λ	k							
1	5	0.001479	0.001436	0.001535	0.000929	0.001503	0.001929	**0.002839**
	10	0.001176	0.001400	0.001569	0.001431	0.001262	0.000925	**0.003606***
	50	0.001176	0.001039	0.001657	0.000769	0.001415	0.001334	**0.008186***
	100	0.001169	0.001137	0.001042	0.000530*	0.001536	0.000982	**0.008331***
10	5	0.001562	0.002907*	0.003969*	0.001780	0.003123*	0.003418*	**0.008163***
	10	0.001642	0.002767*	0.003391*	0.003526*	0.004016*	0.003400*	**0.007680***
	50	0.001454	0.003432*	0.004029*	0.001208*	0.003849*	0.003281*	**0.007949***
	100	0.001419	0.002211*	0.001940*	0.000658*	0.001517	0.001359	**0.008260***
100	5	0.001732	0.003986*	0.003637*	0.001923	0.003866*	0.003487*	**0.007935***
	10	0.001886	0.004396*	0.004920*	0.003619*	0.005861*	0.005805*	**0.008374***
	50	0.002022	0.003388*	0.004083*	0.002638*	0.004501*	0.005079*	**0.007971***
	100	0.001891	0.003191*	0.003044*	0.000670*	0.002738*	0.003062*	**0.008241***
150	5	0.001657	0.006773*	0.007885*	0.001967*	**0.009253***	0.008709*	0.008035*
	10	0.001856	0.004569*	0.003312*	0.003627*	0.005285*	0.005269*	**0.007744***
	50	0.001960	0.005775*	0.004491*	0.001452*	0.004570*	0.003758*	**0.008137***
	100	0.001881	0.002783*	0.002233	0.000712*	0.002502	0.002432	**0.008060***
200	5	0.003324	0.007404*	0.006192*	0.002068*	0.008022*	0.006149*	**0.008220***
	10	0.003112	0.003838*	0.003433	0.003883*	0.003373	0.004230*	**0.006908***
	50	0.002447	0.003543*	0.005781*	0.002908	0.004549*	0.006557*	**0.007958***
	100	0.002288	0.002955*	0.003190*	0.000650*	0.002636*	0.003126*	**0.008105***

Table 2. Comparing the results ($MAP@10$) of our proposal, with the CAMF contextual recommender algorithm using contexts extracted by the *RulesContext* technique, against the results of the baseline MF.

Parameters		Baseline	50_5	50_10	50_20	75_5	75_10	75_20
λ	k							
1	5	0.001479	0.001641	**0.002433**	0.002041	0.001886	0.002178	0.001942
	10	0.001176	**0.002079***	0.001764*	0.001864	0.001581	0.001657	0.002057*
	50	0.001176	0.001614*	**0.002134***	0.001707	0.001471*	0.001984*	0.001924*
	100	0.001169	0.002140*	0.002098*	**0.002295***	0.001458	0.001529	0.001698
10	5	0.001562	0.008969*	0.011458*	**0.011627***	0.003281*	0.008474*	0.008162*
	10	0.001642	0.004001*	0.005019*	0.005521*	**0.006199***	0.004761*	0.003334*
	50	0.001454	**0.004025***	0.002828*	0.000944*	0.003475*	0.003618*	0.001911
	100	0.001419	**0.003026***	0.001507	0.001386	0.002332*	0.001502	0.001591
100	5	0.001732	0.001389	0.000890*	**0.003739***	0.002550	0.001661	0.001482
	10	0.001886	0.004898*	**0.007647***	0.007017*	0.003827*	0.004673*	0.005663*
	50	0.002022	0.007193*	0.007869*	0.008437*	0.005692*	0.009264*	**0.010544***
	100	0.001891	0.002361*	**0.002873***	0.001403*	0.001859	0.001420*	0.001496*
150	5	0.001657	**0.011691***	0.011434*	0.006454*	0.010267*	0.011105*	0.010705*
	10	0.001856	0.005591*	**0.005994***	0.005518*	0.003927*	0.003966*	0.004469*
	50	0.001960	0.001812	**0.002626***	0.002181	0.001459*	0.000776*	0.000517*
	100	0.001881	**0.002637***	0.002622*	0.001840*	0.001663*	0.001429*	0.001506
200	5	0.003324	0.001422*	0.001143*	**0.003600**	0.001795	0.001665*	0.001594*
	10	0.003112	0.005928*	**0.007412***	0.007184*	0.003948*	0.004515*	0.006337*
	50	0.002447	0.007175*	0.007511*	**0.011850***	0.005883*	0.002342	0.011005*
	100	0.002288	0.002799	**0.002874***	0.002874	0.001696	0.001375*	0.001661*

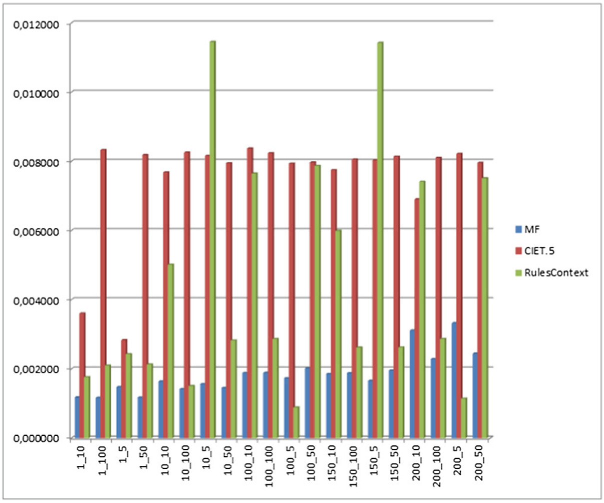

Fig. 4. Comparing the $MAP@10$ values between $CIET.5_{embed}$ and *RulesContext* techniques.

5 Conclusion and Future Work

In this work, we proposed a context-aware recommender method based on text mining (**CARM-TM**). Our method uses the context extraction techniques $CIET.5_{embed}$, which is based on word embeddings; and *RulesContext*, which is based on association rules. For this work, our proposal used the CAMF contextual algorithm to generate the contextual recommendations.

The evaluation was conducted by using the Yelp dataset and the uncontextual MF algorithm as baseline. Our method provided better results than the baseline in all cases. Using the $CIET.5_{embed}$ technique, we obtained good results in most cases. However, the best $MAP@10$ value was provided using the context 50_10 extracted by the *RulesContext* technique.

As future work, we will evaluate our proposal with other datasets and context-aware recommender systems. In addition, we will combine both context extraction techniques, $CIET.5_{embed}$ and *RulesContext*. We will also work on the proposal of a new method for context extraction by using opinion mining.

Acknowledgment. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq) - grants #403648/2016-5 and #426663/2018-7, and Fundação Araucária de Apoio ao Desenvolvimento Científico e Tecnológico do Estado do Paraná - Brasil (FAPPR). The authors also would like to thank FAPESP (grant #2018/04651-0, São Paulo Research Foundation (FAPESP)).

References

1. Aggarwal, C.C.: Recommender Systems. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29659-3
2. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. SIGMOD Rec. **22**(2), 207–216 (1993). https://doi.org/10.1145/170036.170072. http://doi-acm-org.ez67.periodicos.capes.gov.br/10.1145/170036.170072
3. Bauman, K., Tuzhilin, A.: Discovering contextual information from user reviews for recommendation purposes. In: CBRecSys 2014: Proceedings of Workshop on New Trends in Content-based Recommender Systems, pp. 2–9 (2014)
4. Bottou, L.: Stochastic gradient descent tricks. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade. LNCS, vol. 7700, pp. 421–436. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8_25
5. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI 1998: Proceedings of the Fourteenth Conference on Uncertainty in AI, pp. 43–52 (1998)
6. Chen, G., Chen, L.: Recommendation based on contextual opinions. In: Dimitrova, V., Kuflik, T., Chin, D., Ricci, F., Dolog, P., Houben, G.-J. (eds.) UMAP 2014. LNCS, vol. 8538, pp. 61–73. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08786-3_6
7. Chen, L., Chen, G., Wang, F.: Recommender systems based on user reviews: the state of the art. User Model. User-Adapt. Interact. **25**(2), 99–154 (2015)
8. Dey, A.K.: Understanding and using context. Person. Ubiquit. Comput. **5**(1), 4–7 (2001)
9. Hariri, N., Mobasher, B., Burke, R., Zheng, Y.: Context-aware recommendation based on review mining. In: ITWP 2011: Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems, pp. 30–36 (2011)
10. Kim, D., Park, C., Oh, J., Lee, S., Yu, H.: Convolutional matrix factorization for document context-aware recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys 2016, pp. 233–240. ACM, New York (2016). https://doi.org/10.1145/2959100.2959165. http://doi-acm-org.ez67.periodicos.capes.gov.br/10.1145/2959100.2959165
11. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 447–456. ACM, New York (2009)
12. Koren, Y., Bell, R., Volinsky, C., et al.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
13. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems, vol. 15. SIAM, Philadelphia (1995)
14. Li, Y., Nie, J., Zhang, Y.: Contextual recommendation based on text mining. In: COLING 2010: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 692–700 (2010)
15. Sulthana, A.R., Ramasamy, S.: Ontology and context based recommendation system using neuro-fuzzy classification. Comput. Elect. Engin. (2018). https://doi.org/10.1016/j.compeleceng.2018.01.034. http://www.sciencedirect.com/science/article/pii/S0045790617337382
16. Sundermann, C., Antunes, J., Domingues, M., Rezende, S.: Exploration of word embedding model to improve context-aware recommender systems. In: 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 383–388, December 2018. https://doi.org/10.1109/WI.2018.00-64