Proceedings of the Federated Conference on Computer Science and Information Systems pp. 421–428
DOI: 10.15439/2016F141
ACSIS, Vol. 8. ISSN 2300-5963

Predicting Star Ratings based on Annotated Reviews of Mobile Apps

Dagmar Monett and Hermann Stolte
Computer Science Department
Faculty of Cooperative Studies
Berlin School of Economics and Law, Germany
Email: {dagmar.monett-diaz, hermann.stolte}@hwr-berlin.de

***Abstract*—This paper presents and evaluates different computational models for review rating prediction. The models rely solely on star ratings from an annotated corpus of customer reviews of mobile apps that were collected from the Google Play Store in a related work. Fine-granular opinions and the classification of their sentiment orientation were already available. The models build upon them to make predictions based on their polarity. Predicting star ratings is of importance to the sentiment analysis community because it can better be understood how customers subjectively rate products. Rating them consistently with corresponding written reviews, however, remains a difficult task for automated predictors. This paper sheds new light in that direction.**

***Index Terms*—Mobile apps, review rating prediction, semantic sentiment analysis.**

I. Introduction

MOBILE app star ratings and reviews drive apps' rankings, downloads, updates, and in-app purchases. That is what a study from Apptentive has found after surveying smartphone owners and after analysing "historical data from delivering over 160 million interactions and ratings prompts" [1]. According to the study, both star ratings and reviews strongly influence not only the success of mobile apps but also the consumers' engagement with them.

The analysis and interpretation of mobile app star ratings and reviews are not straightforward tasks, however. Monitoring star ratings and reviews is expensive, difficult to accomplish, laborious, and error-prone [2]; many of the ratings and reviews are biased (e.g. app users are more likely to leave ratings or reviews after a negative experience with the app [1]); reviews are in general short and often use abbreviations, emoticons, and informal language; and even star ratings are sometimes unrelated to the experiences with the app itself (e.g. [3] analyses how people give poor ratings just because they are asked to rate the app, explicitly).

Star ratings and reviews are extremely important for brands, for example, for improving their products based on customers' feedback. Ratings also matter for marketing purposes and companies' reputation: it is not only crucial that a top app is highly rated but also that it has at least four stars and many ratings. According to Walz [4], 88% of top-100 Android apps (51% of top-100 iOS apps) have a rating greater than four stars, and the average top-100 Android app (top-100 iOS app) has 3.1 million (196 thousand) ratings.[1] But how to predict or to influence users' star ratings?

Star ratings and reviews are also crucial for customers and their future behaviour when using and recommending the apps. If new customers trust an app's ratings and reviews, then they are more willing to download the app and to benefit from its functionality, e.g. to buy products easily, or to connect and communicate instantly with others, or to simplify daily activities at the office, to name a few benefits. If their experiences with the app are positive, then they would recommend it further and even give feedback to the company for improvements to the app: a win-win situation. Although customers and companies value feedback differently [5], it is true that not only star ratings but also the reviews' content play an important role for both parts.

However, could we *teach* users how to rate apps *consistently* with the review they are writing for a mobile app? For example, would it be possible to improve recommendation accuracy by suggesting to users the most adequate star rating they should give to a product depending on the semantic orientation of what they have already written in the review? How does it compare to previously reviewed mobile apps? Would an improvement in the accuracy also mean an improvement of users' engagement and satisfaction with the apps?

The remaining sections of this paper continue as follows: Section II introduces both the task of review rating prediction and related work in this area. A corpus of annotated reviews of mobile apps from different domains that is used for analysis is presented in Section III. Computational models that are proposed to predict star ratings based on the annotated reviews of the corpus are topic of Section IV. These models are analysed and evaluated in several experimental settings that are defined in Section V. Finally, results are discussed before the conclusions of the paper are presented together with some ideas for further work.

II. Related Work

The prediction of star ratings (e.g., ratings ranging from 1 to 5 stars) has been the focus of many academic and business applications to date. In particular, *review rating prediction*,

[1] Figures from June 2015.

978-83-60810-90-3/$25.00©2016, IEEE

also known as *sentiment rating prediction*, is a task that deals with the inference of an author's implied numerical rating, i.e. on the prediction of a rating score, from a given written review [6], [7]. Recommendation systems, for instance, often suggest products based on star ratings of similar products previously rated by other users.

Yet analysing a textual review is a much more difficult task than guessing the rating by only considering other available numerical scores. This is why not only classifying sentiment [8], [9] but also predicting rating scores has captured the attention of the sentiment analysis community in the last few years. For example, Pang and Lee apply classification and regression, supervised learning techniques to rate movie reviews [10], and Goldberg and Zhu extend their approach by applying a graph-based semi-supervised learning algorithm that achieves better performance [11]. Tang and co-authors follow a similar approach [12], and present a neural network-based method that considers not only the review texts but also author information. They claim that their method "performs better than several strong baseline methods which only use textual semantics." Li and co-authors go beyond the review texts and their authors, and add information also about the product that is reviewed, by modelling all three features using a three-dimensional tensor [13]. Then, they apply tensor factorisation techniques and optimise their model using gradient descent. Their results outperform other similar approaches. Furthermore, Qu et al. introduce the *bag-of-opinions* representation for which their method learns rating scores from domain-independent corpora using constrained ridge regression [14].

Zhang and co-authors delve deeper into the polarity[2] of a review by stating that "it might not be appropriate to use overall ratings as ground-truth to label the sentiment orientations of review texts, as users tend to act differently when making overall ratings and expressing their true feelings on detailed product aspects or features" [15]. This means that rating predictors should consider the subtle differences between review texts as a whole, and reviews of individual aspects. [16] and [17] come to the same conclusions, and affirm that textually derived ratings are better predictors than numerical star ratings. In their experiments, Zhang and co-authors first let three annotators manually label the polarity orientation of sample reviews from a restaurant dataset and then compare them against automatically generated annotations using unsupervised review-level sentiment classification [15]. Afterwards, the annotators label not reviews as a whole but their aspects or features individually. Again, the results are compared to those obtained with the methods the authors propose, showing the inconsistency between textual reviews and numerical ratings when the latter do not consider phrase-level sentiment polarity.

Gupta and co-authors also apply supervised learning with a multi-aspect rating prediction for textual reviews of restaurants [18]. They consider numerical ratings for aspects like food, service, and overall experience, inter alia, as well as considering the interdependence of aspects for around eight sentences per review on average. Orimaye and co-authors introduce a sentence-level polarity correction [19]. Their technique identifies sentences with inconsistent polarities that are handled as outliers and, as such, are discarded from the reviews. This approach might not be convenient for mobile app reviews, where the length of subjective phrases might be about two words long on average, and the reviews are not long enough either [20]. Discarding information in the case of mobile apps would introduce an extra bias to the problem.

Sänger [20] introduces an aspect-based opinion mining of mobile apps ratings that extends Klinger and Cimiano's work [21], [22]. According to Sänger, Klinger and Cimiano's approach was chosen because it deals with fine-granular aspect-based opinion mining, its implementation is open-sourced (see https://bitbucket.org/rklinger/jfsa), and it is suitable for mining text written in German, as is the case of the dataset he uses (see next section). Sänger concludes that such a technique is also appropriate for analysing mobile app reviews; he both adapts and validates Klinger and Cimiano's work for such reviews.

Sänger's approach serves as the background to, and the basis for, the work presented here. It is worth mentioning, however, that the goal of the work presented in this paper *is not* to deal with aspect identification nor with sentiment classification; but assuming that these tasks are performed *before* the star ratings are predicted. A complement to Sänger's work, in other words. Thus, *unlike* other approaches that identify aspects or classify sentiment at a fine-granular level, like most of the works reviewed above (e.g. [10]–[12], [17], [21], to cite but a few), the idea of our approach is to provide a method for predicting star ratings based *solely* on available annotated, fine-granular opinions.

The next section introduces the dataset that is used for analysis and validation.

III. Corpus of Annotated Customer Reviews

The annotated corpus used here was initially provided by Sänger as constructed in [20], later named *SCARE* as introduced in [23]. It consists of 1,760 randomly selected, annotated reviews for a total of 130 mobile apps from different domains. The annotations consider fine-granular opinions as well as the app aspects and their relationships. Each textual review includes a customer evaluation of the app, and has an associated rating. All textual reviews are in German. Each evaluation consists of at least one phrase. There is a total of 6,446 phrases from which 3,959 are manually annotated subjective phrases. The corpus contains a total of 2,487 aspects.

Sänger claims that his mobile app dataset is the first of its kind. It comprises a total of 802,860 reviews in German of 148 mobile apps from 11 different categories, the annotated corpus introduced above being a subset of it. The reviews were collected from the Google Play Store (see https://play.google.

[2] See next section for more on polarity.

com/store) using an open-source API for the Android Market (see https://code.google.com/archive/p/android-market-api/).

Specifically, the annotated corpus that is used here contains the following information, which follows the structure presented in [24]:

all.data A list of all reviews as retrieved through the Android Market API, including the app's name, the full review text, and the star rating given by the user.

all.txt A list of all review texts as they were used in the annotation process (the review title and its content are concatenated).

all.csv A list of all annotated subjective phrases and aspects, each subjective phrase with an internal ID, its corresponding ID, and its polarity.

all.rel A list of all annotated relations between the subjective phrases and their aspects.

Figure 1 shows all major steps of the prediction process that makes use of the annotated corpus. It starts by *parsing* the lists introduced above and by creating workspace variables with which to work.

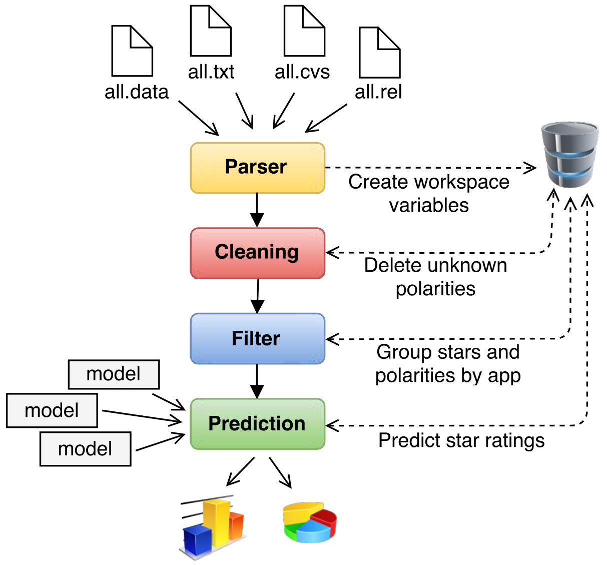

Fig. 1 Prediction process.

The polarity of a phrase depends on the expressed opinion, and thus on the semantic orientation or *sentiment* of the phrase, i.e., whether the expressed opinion of the opinion holder[3] is positive, negative, or neutral [7]. Since a review might have more than one phrase, calculating the polarity of the review would depend on the polarities of its phrases. In particular, mobile app reviews are much shorter than other product reviews, use language constructs that are similar to those used in micro-blogging (e.g., Twitter), have unstructured sentences in general, and often use more concise words [20].

According to Sänger [20], the subjective phrases were annotated and their polarity determined following a rigorous process that comprised the development of annotation guidelines, the explicit training of four annotators on these guidelines, the annotation of random phrases in iterative rounds, as well as a later controlling and improvement of the performed annotations. The final version of the annotations during the training process achieved a substantial inter-annotator agreement with a kappa value $\kappa = 0.72$, computed using the Fleiss' kappa measure (see Chapter 3 in [20] for more). Then, the actual annotations to be considered for the corpus were carried out.

It is worth mentioning that the polarity of type *unknown* was handled as a default value in the tool that was used for annotating the corpus (see http://brat.nlplab.org/). According to Sänger [25], this relates to reviews where the annotators forgot to specify the polarities. Because the phrase polarity was not of further interest in his work, there was no need to correct that issue. Thus, unknown sentiments are not considered for the experiments that will be introduced in succeeding sections: they are deleted from the corpus in a *cleaning* procedure (see Figure 1).

After cleaning the unknown polarities out, the new annotated corpus consists of 1,751 reviews, 130 apps, 6,398 phrases, and 3,927 subjective phrases. Table I shows the distribution of all phrases from the corpus according to their polarity, before and after the cleaning process has taken place. Almost two thirds of the subjective phrases express a positive opinion, and about one-third have a negative polarity.

TABLE I
POLARITY DISTRIBUTION OF ANNOTATED SUBJECTIVE PHRASES.

	Before cleaning		After cleaning	
Polarity	Annotated phrases	%	Annotated phrases	%
positive	2,463	62.2	2,458	62.6
negative	1,433	36.2	1,416	36.1
neutral	53	0.01	53	1.3
unknown	10	0.002	–	–

The star ratings associated with the entries from the corpus, i.e., to the annotated mobile apps reviews, after the cleaning process are summarised in Table II.

TABLE II
DISTRIBUTION OF STAR RATINGS.

Star rating	No. of annotated reviews	%
1	295	16.8
2	111	6.3
3	136	7.8
4	299	17.1
5	910	52.0

If reviews with 4-5 stars are considered positive reviews and those with 1-2 stars are considered negative reviews (the *thumbs-up-thumbs-down* approach suggested by Liu in [7]), then over two-thirds of the reviews from the annotated corpus have a positive polarity (69.1%) and only about one out of four reviews is negative (23.1%). Compared to the

[3] The person that holds the opinion [9]. Also, opinion source.

subjective phrases polarities from Table I, these are slightly smaller values (62.6% positive polarity). This means that the expressed opinions from the corpus are in general more positive when they are given as an overall numerical rating than when taking into account their individual subjective phrases (probably aspect-related) polarity. It can be observed in Figure 2 that the line depicting the average of star ratings is above the expected line averaging the subjective phrases polarity. The fine-granular analysis suggested by Klinger and Cimiano [21], [22] and extended by Sänger [20] confirms the findings from other approaches [15]–[17] with respect to the subtle differences between ratings of reviews as a whole and as differentiated subjective phrases.

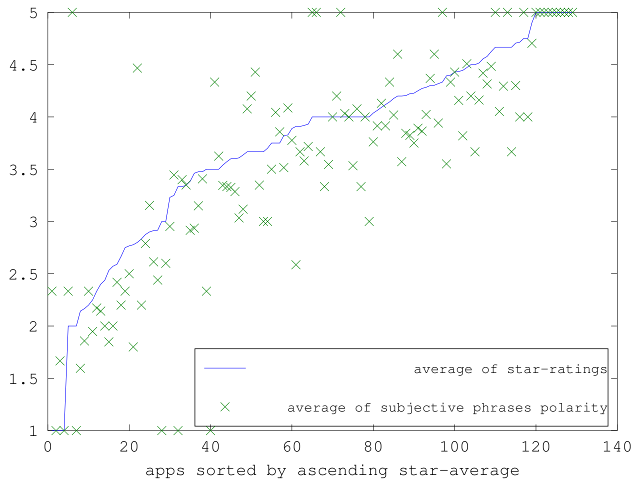

Fig. 2 Average of labelled star ratings versus average of subjective phrases polarity.

Figure 3 shows the number of star ratings and subjective phrases for each app after a *filtering* procedure (see Figure 1) that groups them together according to the reviews associated with that app. There are about twice as many subjective phrases than star ratings per app. They have a strong linear dependency: there is a positive correlation, with the Pearson's correlation coefficient $\rho = 0.8$. This is a good indicator for considering linear regression models that can predict the star ratings without human intervention.

Yet another possibility to plot the data from the corpus is shown in Figure 4. This time, the number of reviews per app is taken into account. Such a visualisation was helpful when analysing apps according to their importance or to the number of reviews that are provided. We do not consider further implications in our experiments but were better aware of the distribution of the ratings when analysing the data.

Not only is a visual analysis of the data concerning the number of reviews and their ratings interesting, but also in which relation stay positive and negative opinions to each other. As can be seen in Figure 5, negative reviews have higher impact than positive reviews. There is a negative correlation between both of them, with Pearson's correlation coefficient $\rho = -0.78$ (apps with no positive subjective phrases were filtered out to avoid indetermination when considering the negative vs. positive sentiment ratio).

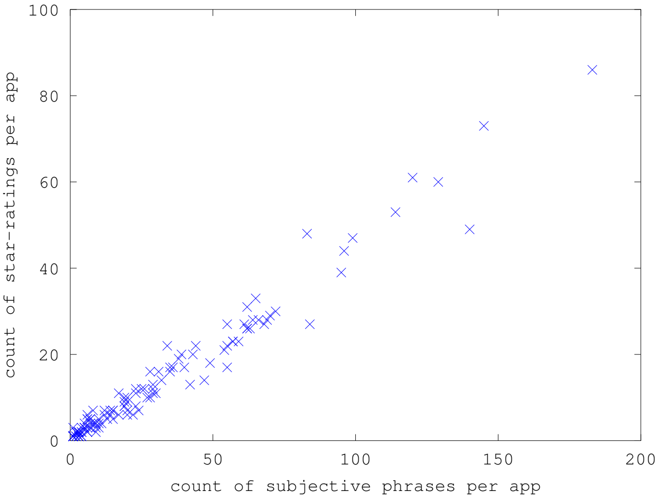

Fig. 3 Number of star ratings and subjective phrases for each app in the annotated corpus.

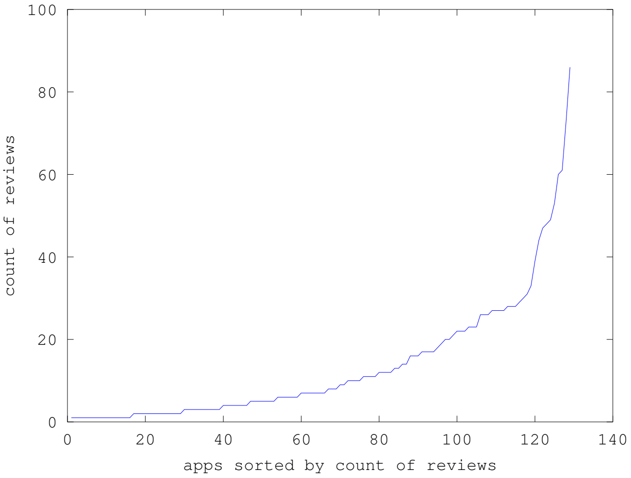

Fig. 4 Count of reviews per app sorted in ascending order.

All these observations determined which computational models should be considered in order to predict star ratings. Some of these models will be presented in the next section. In this paper and for the reasons commented above (like the strong linear dependency between subjective phrases and star ratings per app), we place great emphasis on multivariate regression models.

IV. Prediction of Star Ratings

Let $h_\Theta : \mathbb{R}^{n+1} \to \mathbb{R}$ be the hypothesis of a multivariate regression model,

$$h_\Theta(\mathbf{x}) = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n = \Theta^\intercal \mathbf{x}, \qquad (1)$$

with $\Theta \in \mathbb{R}^{n+1}$ being a vector of parameters, $\mathbf{x} \in \mathbb{R}^{n+1}$ being a vector of features or independent variables, $n \in \mathbb{N}$, and $i = 0, \ldots, n$.

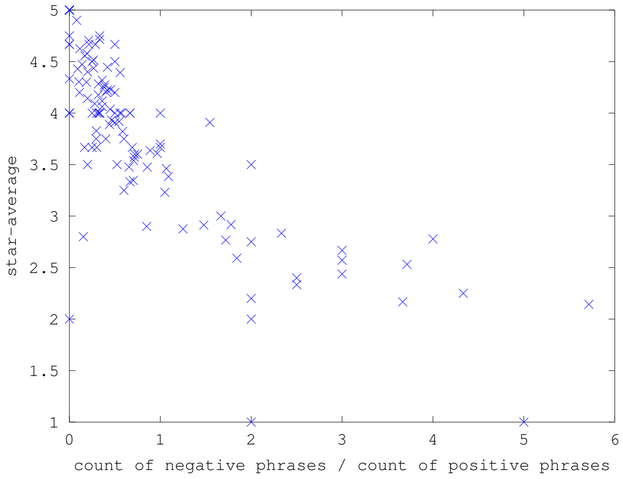

Fig. 5 Star average according to the negative vs. positive sentiment ratio.

The cost function which is to be minimized in order to find the optimal values of the parameters θ_i is the following:

$$c(\Theta) = \frac{1}{2m}\sum_{j=1}^{m}(h_\Theta(\mathbf{x})^{(j)} - y^{(j)})^2, \tag{2}$$

with $m = 1701$ being the number of reviews in the corpus and y being the dependent variable or star rating for each annotated review j.

For predicting star ratings of mobile apps, a model with four variables (or features) could be considered, where

- x_0 is equal to 1 for convenience of notation,
- x_1 is the number of subjective phrases with positive polarity,
- x_2 is the number of subjective phrases with negative polarity, and
- x_3 is the number of subjective phrases with neutral polarity.

Let $h_{1_\Theta} : \mathbb{R}^4 \to \mathbb{R}$ be the corresponding hypothesis:

$$h_{1_\Theta}(\mathbf{x}) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3. \tag{3}$$

This is the *baseline model*.

In case the neutral polarities are not considered, as will be discussed in the next section, the above model can be simplified as follows:

$$h_{2_\Theta}(\mathbf{x}) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2, \tag{4}$$

with $h_{2_\Theta} : \mathbb{R}^3 \to \mathbb{R}$.

Since some apps might have many more reviews than others, the values of the features could be normalised using the following scaling:

$$x'_i = \frac{x_i - \mu_i}{\sigma_i}, \tag{5}$$

with μ_i the mean value and σ_i the standard deviation of the feature i in the vector of features $\mathbf{x}$.

An average-based, simpler model could also be considered by taking into account only the average value of the polarities of a review (e.g., average polarity between all positive, negative, and neutral sentiments of the review) in one feature. Let $h_{3_\Theta} : \mathbb{R}^2 \to \mathbb{R}$ be the hypothesis for that case:

$$h_{3_\Theta}(\mathbf{x}) = \theta_0 x_0 + \theta_1 x_1. \tag{6}$$

The average polarity (numerical) value of a review can be calculated by mapping the polarities to the following values: 5 for a positive polarity, 3 for a neutral polarity, and 1 for a negative polarity.

The average polarity value can also be calculated by considering the *review rating score* (RRS) as suggested in [16] and [17]. This would mean that only the positive and negative polarities are taken into account, and are summed up using the following formula:

$$RRS^{(j)} = \left(\frac{P^{(j)}}{P^{(j)} + N^{(j)}} \cdot 4\right) + 1, \tag{7}$$

where $P^{(j)}$ is the number of positive subjective phrases in review j, $N^{(j)}$ is the number of negative subjective phrases, and $1 \leq j \leq m$. As in [16], the new rating is scaled in the range of the corpus star rating (i.e., one to five stars).

Even a polarity ratio can be computed, too, where only the proportion between negative and positive polarities is taken into account.

Altogether, eight different models will be analysed and evaluated in the experiments that are introduced in the next section. They are summarised in Table III.

TABLE III Overview of prediction models.

Model	Hypothesis	Neutral polarities	Features normalised	Polarity average	RSS average	Polarity ratio
M1	h_{1_Θ}	✓				
M2	h_{1_Θ}	✓	✓			
M3	h_{2_Θ}					
M4	h_{2_Θ}		✓			
M5	h_{3_Θ}	✓		✓		
M6	h_{3_Θ}			✓		
M7	h_{3_Θ}				✓	
M8	h_{3_Θ}					✓

V. Experiments

Two different groups of experiments are considered for predicting the star ratings of mobile apps based on the expressed opinions from each review. All rely only on the polarity of the subjective phrases that are included in the annotated corpus.

The first group of experiments deals with assessing the importance of sentiment in the reviews. For example, whether to filter *neutral* phrases out from the corpus or not is investigated by applying different regression models, as introduced in the section above. Furthermore, filtering reviews with no sentiment out (i.e., those that do not contain subjective phrases at all) is also analysed.

The second group of experiments makes use of other predictors, as suggested in [16] and [17], after considering the results of the first group of experiments.

Each individual experiment is run 10,000 times. A Monte Carlo cross-validation[4] is applied each time: on each iteration, the annotated reviews dataset is randomly partitioned into a 70% training dataset that is used to train the model in a supervised manner, and into a 30% testing dataset that is used to validate it.

A. Multi-variate linear regression-based predictors

Ganu et al. point out that neutral polarities do not add significant information to their experiments [16]. This could be also the case for the sentiment rating prediction of mobile apps that are used here. In order to investigate this, some of the regression models introduced in Section IV are trained and evaluated both with and without taking into account the neutral sentiments. Furthermore, they are also trained and evaluated with a reduced corpus that does not contain reviews that have no subjective phrases at all, i.e., reviews with no positive, neutral, or negative phrases are filtered out from the corpus. Concretely, a total of 77 reviews are filtered out.

The first experiment, experiment *E1*, considers the polarity count and evaluates the baseline regression model, i.e., hypothesis $h_{1_\Theta}(\mathbf{x})$ from Equation 3 and hypothesis $h_{2_\Theta}(\mathbf{x})$ from Equation 4. In other words, models *M1* and *M3* are evaluated, i.e., with and without neutral polarities.

The second experiment, experiment *E2*, uses the average-based hypothesis $h_{3_\Theta}(\mathbf{x})$ from Equation 6 for the training. Models *M5* and *M6* are evaluated, i.e., with and without neutral polarities.

The third experiment, experiment *E3*, considers the baseline model and the model without neutral polarities, i.e., hypotheses $h_{1_\Theta}(\mathbf{x})$ and $h_{2_\Theta}(\mathbf{x})$, both with normalised features. Models *M2* and *M4* are evaluated.

B. Univariate, average-based predictors

This group of experiments considers the RRS as defined in Equation 7.

First, an experiment *E4* with hypothesis $h_{3_\Theta}(\mathbf{x})$ is considered. In this case, model *M7* is evaluated.

A second experiment, experiment *E5*, also uses hypothesis $h_{3_\Theta}(\mathbf{x})$ but with the negative vs. positive polarities ratio, i.e., model *M8* is evaluated.

A third experiment, experiment *E7*, makes a *metadata-based prediction* (also similar to that proposed in [16]): given a new test review of an app, it predicts the rating by computing the average of all reviews available in the training set. A hypothesis like that from Equation 6 is considered and, with it, a new model *M9* is evaluated.

VI. Results and Discussion

The results show the averages of the mean squared errors (MSE) and the standard deviation σ for both the training and the test sets for each of the 10,000 iterations from the experiments. Together with these metrics, the value of the maximum error minus the minimum is also given.

Table IV shows the results for the first group of experiments, i.e., for those settings that evaluate not only the importance of neutral sentiment orientation but also whether reviews without subjective phrases should be included in the analysis or not.

The model that best predicts the star ratings is *M6* (see the last column of experiment *E2* in Table IV). This means that filtering both subjective phrases with neutral polarity *and* reviews with no sentiment orientation at all, fits much better the predictor (i.e., hypothesis $h_{3_\Theta}(\mathbf{x})$ from Equation 6) to the observed data.

In a second grade of importance are the best results that were obtained for experiments *E1* and *E3*. These are underlined. For our concrete corpus, it is not a good idea to normalise the model features: this does not improve the accuracy (see the second-last column of experiment *E3* in Table IV). Furthermore, models with more features profit from more data, as expected (see the first column of experiment *E1* in Table IV).

Figure 6 shows a visual comparison between the results of the first two experiments, *E1* and *E2*.

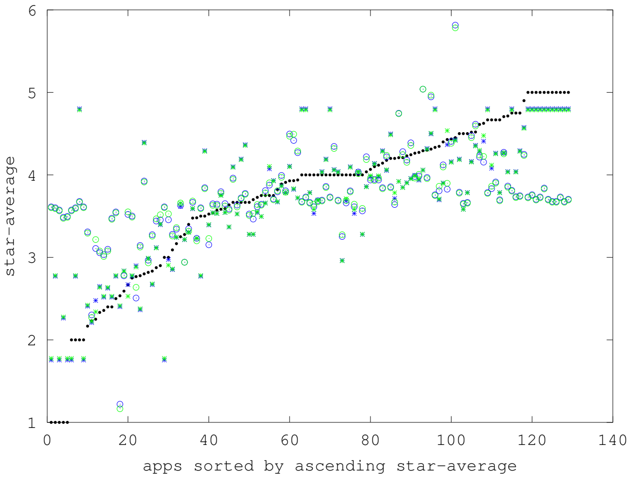

Fig. 6 Rating prediction for experiments *E1* and *E2*. Black asterisks: labels, blue (dark gray) circles: *E1* with neutral phrases, blue (dark gray) asterisks: *E2* with neutral phrases, green (light gray) circles: *E1* without neutral phrases, green (light gray) asterisks: *E2* without neutral phrases.

Since the hypothesis of the best model so far is h_{3_Θ}, then predicting the star rating for a new app given its review[5] would mean evaluating the hypothesis as follows:

$$h_{3_\Theta}(\mathbf{x}) = 1.0814 + 0.73538x_1,$$

[4] The Monte Carlo cross-validation is a non-exhaustive cross-validation technique.

[5] And after having classified the sentiment orientation of its subjective phrases.

TABLE IV Mobile apps rating prediction: Importance of sentiment in the reviews.

Experiments	with neutral phrases						without neutral phrases					
	with reviews with no subjective phrases			without reviews with no subjective phrases			with reviews with no subjective phrases			without reviews with no subjective phrases		
	MSE	σ	max-min	MSE	σ	max-min	MSE	σ	max-min	MSE	σ	max-min
E1: Linear regression with polarity count												
Training	0.60971	0.07354	0.51094	0.68953	0.08169	0.57188	0.60967	0.07364	0.50569	0.69099	0.08109	0.59013
Test	0.67642	0.17841	1.25400	0.75563	0.19703	1.57250	0.67660	0.17898	1.32320	0.75187	0.19591	1.49400
E2: Linear regression with polarity average												
Training	0.29072	0.04762	0.28186	0.26790	0.04754	0.25595	0.29048	0.04842	0.28524	**0.26720**	**0.04739**	**0.24608**
Test	0.31143	0.11380	0.72518	**0.28208**	0.11246	0.66817	0.31222	0.11589	0.69722	0.28359	**0.11206**	**0.59792**
E3: Linear regression with normalized polarity count												
Training	0.61055	0.07457	0.53986	0.68973	0.08230	0.60612	0.61063	0.07440	0.53547	0.68895	0.08144	0.57357
Test	0.67493	0.18186	1.48680	0.75434	0.19820	1.65960	0.67396	0.18094	1.35310	0.75660	0.19710	1.50860

where x_1 is the average of the positive and negative polarities of the review, and the intercept and the slope are the optimal parameters Θ that were found.

Table V shows the results for the second group of experiments.

TABLE V Mobile apps rating prediction: Other (average-based) predictors.

Experiments	with neutral phrases			without neutral phrases		
	MSE	σ	max-min	MSE	σ	max-min
E4: Linear regression with RRS						
Training	–	–	–	**0.23979**	**0.04484**	**0.21852**
Test	–	–	–	**0.25547**	**0.10604**	**0.50679**
E5: Linear regression with ratio neg/pos polarities						
Training	0.79105	0.08650	0.59050	0.87870	0.09371	0.65414
Test	0.82057	0.20284	1.55290	0.91346	0.21984	1.63780
E6: metadata-based prediction						
Training	–	–	–	–	–	–
Test	–	–	–	2.35960	0.08397	0.63069

If the review rating score is considered, i.e., model *M7*, then its results outperform all other predictions (see the final column of experiment *E4* in Table V).

Figure 7 shows a closer look when comparing the best models of both groups of experiments, i.e., *E2* and *E4*.

The predictions that are computed based on the review rating score are much closer to the star ratings given by the authors of the reviews, as Figure 8 clearly indicates (compared to those of Figure 2).

VII. Conclusions

Textually-derived rating prediction can be performed well even when only phrase-level sentiment polarity is available. This is what the computational models introduced and evaluated in this paper have shown. Not all fine-granular opinions are of importance, however: filtering out subjective phrases with neutral sentiment and computing the overall sentiment of a review using the review rating score proposed in [16] and [17] provides the best star rating predictions for mobile apps' reviews written in German. Based on these results, new applications could suggest to customers how to

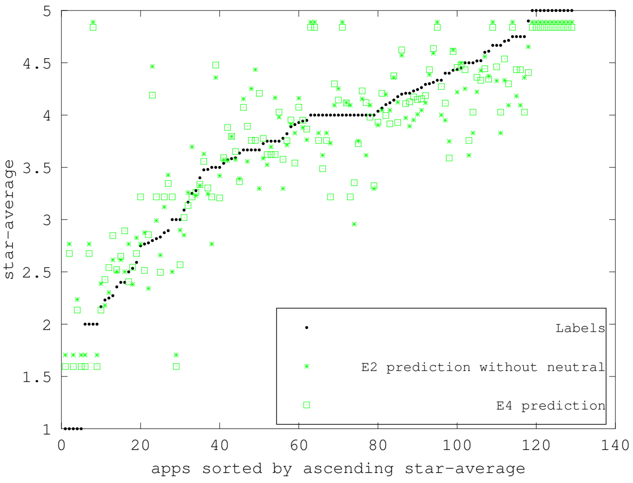

Fig. 7 Rating prediction for experiments *E2* and *E4*. Black dots: labels, green (light gray) asterisks: *E2* without neutral phrases, green (light gray) boxes: *E4* without neutral phrases.

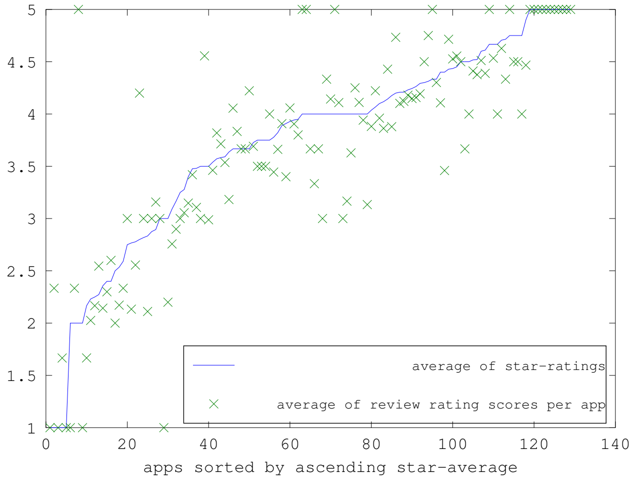

Fig. 8 Review rating scores per app.

rate apps more consistently with the reviews they write, by considering their expressed opinions at the phrase level. Both customers and companies would benefit alike.

Further work will deal with the ideas that follow. Subjective phrases are aspect-oriented, i.e., the expressed opinions are probably related to features or aspects of a particular app. By extending the model to consider the aspects' relevance, an improvement in performance might be achieved. Furthermore, the phrase polarity is usually given in broad categories (i.e. positive, neutral, and negative). It could be interesting to analyse the *strengths* of the opinions [26], too. Moreover, it is our interest dealing with other types of models different than linear, multivariate regression ones.

Acknowledgements

We would like to thank Mario Sänger for insightful discussions and for providing us with the annotated customer mobile apps' reviews dataset.

References

[1] A. Walz and R. Ganguly, *Apptentive 2015 Consumer Survey: The Mobile Marketer's Guide to App Store Ratings & Reviews*. Apptentive, 2015.

[2] Applause, "Listen to the Voice of Your Customers," n.d., available online at http://www.applause.com/resources#whitepapers, retrieved March 13, 2016.

[3] M. Galligan, "The right way to ask users to review your app," 2014, available online at https://medium.com/circa/the-right-way-to-ask-users-to-review-your-app-9a32fd604fca#.kud43shhq, retrieved March 13, 2016.

[4] A. Walz, "Dissecting the App Store Top Charts: The Anatomy of a Top App," 2015, available online at http://www.apptentive.com/blog/app-store-top-charts/, retrieved March 29, 2016.

[5] M. Smith, "Feedback and Loyalty on the Mobile Frontier: New Research From Apptentive and SurveyMonkey," 2016, available online at http://www.apptentive.com/blog/feedback-and-loyalty-on-the-mobile-frontier/, retrieved March 30, 2016.

[6] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[7] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[8] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," in *Proceedings of the* 43^{rd} *Conference on Empirical Methods in Natural Language Processing, EMNLP'02*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 79–86.

[9] B. Liu, "Sentiment Analysis and Subjectivity," in *Handbook of Natural Language Processing*, 2nd ed., N. Indurkhya and F. J. Damerau, Eds. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group, 2010.

[10] B. Pang and L. Lee, "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with respect to Rating Scales," in *Proceedings of the* 43^{rd} *Annual Meeting of the Association for Computational Linguistics, ACL'05*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 115–124.

[11] A. B. Goldberg and X. Zhu, "Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization," in *Proceedings of the* 1^{st} *Workshop on Graph Based Methods for Natural Language Processing, TextGraphs-1'06*.

[12] D. Tang, B. Qin, T. Liu, and Y. Yang, "User Modeling with Neural Network for Review Rating Prediction," in *Proceedings of the* 24^{th} *International Joint Conference on Artificial Intelligence, IJCAI'15*, Q. Yang and M. Wooldridge, Eds. Palo Alto, CA, USA: AAAI Press, 2015, pp. 1340–1346.

[13] F. Li, N. Liu, H. Jin, K. Zhao, Q. Yang, and X. Zhu, "Incorporating Reviewer and Product Information for Review Rating Prediction," in *Proceedings of the* 22^{nd} *International Joint Conference on Artificial Intelligence, IJCAI'11*, T. Walsh, Ed., vol. 3. Menlo Park, CA, USA: AAAI Press, 2011, pp. 1820–1825.

[14] L. Qu, G. Ifrim, and G. Weikum, "The Bag-of-Opinions Method for Review Rating Prediction from Sparse Text Patterns," in *Proceedings of the* 23^{rd} *International Conference on Computational Linguistics, Coling'10*, C.-R. Huang and D. Jurafsky, Eds., vol. 2. Beijing, China: Tsinghua University Press, 2010, pp. 913–921.

[15] Y. Zhang, M. Zhang, Y. Liu, and S. Ma, "Boost Phrase-level Polarity Labelling with Review-level Sentiment Classification," *Computational Linguistics*, vol. 1, no. 1, pp. 1–25, 2006.

[16] G. Ganu, N. Elhadad, and A. Marian, "Beyond the Stars: Improving Rating Predictions Using Review Text Content," in *Proceedings of the* 12^{th} *International Workshop on the Web and Databases, WebDB'09*, 2009, pp. 1–6.

[17] G. Ganu, Y. Kakodkar, and A. Marian, "Improving the Quality of Predictions using Textual Information in Online User Reviews," *Information Systems*, vol. 38, no. 1, pp. 1–15, March 2013.

[18] N. Gupta, G. Di Fabbrizio, and P. Haffner, "Capturing the Stars: Predicting Ratings for Service and Product Reviews," in *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search, SS'10*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 36–43.

[19] S. O. Orimaye, S. M. Alhashmi, E. Siew, and S. J. Kang, "Review-Level Sentiment Classification with Sentence-Level Polarity Correction," *Computer Science, OALib Journal*, pp. 1–15, 2015.

[20] M. Sänger, "Aspektbasierte Meinungsanalyse von Bewertungen mobiler Applikationen," Master Thesis, Computer Science Dept., Humboldt-Universität zu Berlin, Berlin, Germany, December 2015.

[21] R. Klinger and P. Cimiano, "Bi-directional Inter-dependencies of Subjective Expressions and Targets and their Value for a Joint Model," in *Proceedings of the* 51^{st} *Annual Meeting of the Association for Computational Linguistics, ACL'13*, R. Navigli, J.-S. Chang, and S. Faralli, Eds., vol. 2. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 848–854.

[22] ——, "Joint and Pipeline Probabilistic Models for Fine-grained Sentiment Analysis: Extracting Aspects, Subjective Phrases and their Relations," in *Proceedings of the IEEE* 13^{th} *International Conference on Data Mining Workshops, ICDMW'13*, W. Ding, T. Washio, H. Xiong, G. Karypis, B. Thuraisingham, D. Cook, and X. Wu, Eds. Dallas, TX, USA: IEEE Computer Society, December 2013, pp. 937–944.

[23] M. Sänger, U. Leser, S. Kemmerer, P. Adolphs, and R. Klinger, "SCARE – The Sentiment Corpus of App Reviews with Fine-grained Annotations in German," in *Proceedings of the* 10^{th} *International Conference on Language Resources and Evaluation, LREC'16*, N. Calzolari, K. Choukri, T. Declerck, M. Grobelnik, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Paris, France: European Language Resources Association (ELRA), May 2016.

[24] R. Klinger and P. Cimiano, "The USAGE review corpus for fine grained multi lingual opinion analysis," in *Proceedings of the* 9^{th} *International Conference on Language Resources and Evaluation, LREC'14*, N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Reykjavik, Iceland: European Language Resources Association, May 2014, pp. 2211–2218.

[25] M. Sänger, private communication, 2016.

[26] T. Wilson, J. Wiebe, and R. Hwa, "Just how mad are you? Finding strong and weak opinion clauses," in *Proceedings of the* 19^{th} *National Conference on Artificial Intelligence, AAAI'04*, G. Ferguson and D. McGuinness, Eds. Menlo Park, California: AAAI Press, 2004, pp. 761–767.