A Multi-criteria Recommender System Exploiting Aspect-based Sentiment Analysis of Users' Reviews

Cataldo Musto
Dept. of Computer Science - University of Bari
cataldo.musto@uniba.it

Marco de Gemmis
Dept. of Computer Science - University of Bari
marco.degemmis@uniba.it

Giovanni Semeraro
Dept. of Computer Science - University of Bari
giovanni.semeraro@uniba.it

Pasquale Lops
Dept. of Computer Science - University of Bari
pasquale.lops@uniba.it

ABSTRACT

In this paper we propose a *multi-criteria recommender system* based on collaborative filtering (CF) techniques, which exploits the information conveyed by *users' reviews* to provide a multi-faceted representation of users' interests.

To this end, we exploited a framework for *opinion mining* and *sentiment analysis*, which automatically extracts relevant aspects and sentiment scores from users' reviews. As an example, in a *restaurant recommendation* scenario, the aspects may regard *food quality*, *service*, *position*, *athmosphere* of the place and so on. Such a multi-faceted representation of the user is used to feed a multi-criteria CF algorithm which predicts user interest in a particular item and provides her with recommendations.

In the experimental session we evaluated the performance of the algorithm against several state-of-the-art baselines; Results confirmed the insight behind this work, since our approach was able to overcome both single-criteria recommendation algorithms as well as more sophisticated techniques based on matrix factorization.

CCS CONCEPTS

• **Information systems → Recommender systems**;

KEYWORDS

Recommender Systems, Opinion Mining, Sentiment Analysis

ACM Reference format:
Cataldo Musto, Marco de Gemmis, Giovanni Semeraro, and Pasquale Lops. 2017. A Multi-criteria Recommender System Exploiting Aspect-based Sentiment Analysis of Users' Reviews. In *Proceedings of RecSys'17, August 27–31, 2017, Como, Italy, ,* 5 pages.
DOI: http://dx.doi.org/10.1145/3109859.3109905

1 BACKGROUND AND MOTIVATIONS

The massive spread of platforms for sharing opinions and reviews is one of the most interesting trends we are recently witnessing in the context of *Web 2.0*. Given how much data is floating around in the Web, it is quite straightforward to think about new strategies to gather and process such data, since several applications may benefit from such a better picture of what people *think* and what people *like*.

As an example, the exploitation of the information conveyed by users' reviews can be useful to tackle one of the issues Recommender Systems (RS) typically suffer from. Indeed, RSs model users' opinions about a certain item through a single discrete scale. This is a clear over-simplification, since the user who expresses the rating may have enjoyed some aspects of the item (e.g. *the quality of the food*) and may have not enjoyed others, as the *service* of the restaurant. The idea of learning users' preferences [6] by expressing several ratings, in order to evaluate different facets of the items has already been investigated in literature, and those systems fall under the name of *multi-criteria recommender systems* [2, 10]. The first attempts towards this direction [8, 9, 14, 15] showed that multi-criteria ratings can improve the predictive accuracy of the system, but they suffer from the issue of overcharging the users during the training process, by asking them to rate the same item many times.

In this work we tackle this problem by proposing a *multi-criteria RS* which automatically learns the preferences of the target user by mining the information conveyed by the reviews she wrote. To this end, we exploited a framework for *aspect extraction* and *sentiment analysis* which automatically extracts the *aspects* discussed in the review along with their *sentiment*, and we used this multi-faceted representation of users' preferences to feed a neighborhood-based CF algorithm which predicts the interest of the target user in a particular item.

As shown by Chen et al. [5], this class of RS is typically referred to as *review-based RS*. Regardless their effectiveness, such systems typically rely on a fixed set of aspects (e.g., in a restaurant scenario: *food*, *atmosphere*, *position*, *service*) which is statically defined and is automatically filled in by analyzing the review. As an example, Chen et al. [4] defined a set of eight different hotel-related aspects and five restaurant-related ones. Differently from such literature, in our work also the set of the aspects we used to represent the interests of the user is automatically learned from the text, as [13] proposed for Chinese restaurant reviews. Moreover, a distinguishing and novel aspect of this work lies in the ability of extracting also *couple of related aspects* (as *food-meat*, *food-fish*), thus building a more fine-grained representation of the preferences.

The rest of the paper is organized as follows: in Section 2 we describe the framework for aspect extraction and our multi-criteria RS.

RecSys'17, August 27–31, 2017, Como, Italy

ISBN 978-1-4503-4652-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3109859.3109905

```
[Review] "I loved the food, the best
sweet and sour soup ever! Everything
was delicious and favorable. Delivery
is pretty good, nice people."

Overall: 4.0 / 5.0

Aspect          Sentiment score
food            4.0
people          4.0
food-soup       5.0
```

Figure 1: Our framework for aspect extraction at work.

Next, Section 3 shows the findings emerged from the experiments and in Section 4 we define future research directions.

2 METHODOLOGY

In the first part of this section we describe our framework for opinion mining and sentiment analysis, while in the latter we show how aspects and sentiment scores extracted from the reviews are used to feed a multi-criteria collaborative filtering algorithm.

Extracting Aspects and Sentiment from Reviews. In this work we exploit a framework for Sentiment Aspect-Based Retrieval called SABRE, proposed by Caputo et al. [3]. In a nutshell, given a set of reviews[1] $R = \{r_1, r_2 \ldots r_n\}$, SABRE produces as output a set of quintuples $\langle r_i, a_{ij}, a_{ijk}, rel(a_{ijk}, r_i), sent(a_{ijk}, r_i)\rangle$ representing the review r_i, the *j-th* aspect a_{ij} extracted from the review, the *k-th* sub-aspect a_{ijk} related to the aspect a_{ij}, the relevance $rel(a_{ijk}, r_i)$ of the aspect in the document, and its sentiment $sent(a_{ijk}, r_i)$, respectively. In order to carry out the aspect extraction process, we used the *Kullback-Leibler divergence (KL-divergence, referred to as δ)*, a non-symmetric measure of the difference between two distributions. Formally, given two corpora c_a and c_b and a term t, pointwise KL-divergence is calculated as:

$$\delta_t(c_a||c_b) = p(t, c_a) log \frac{p(t, c_a)}{p(t, c_b)} \tag{1}$$

Our algorithm for aspect extraction is based on the idea that the use of language differs when talking about a specific domain with respect to a general topic, then this method aims at selecting the aspects whose distributions in a specific domain (e.g. *restaurant reviews*) diverge from those in a general corpus (e.g. the British National Corpus BNC[2]). Thus, we are interested in those aspects mentioned in the review more often than usual. Our strategy for identifying the main aspects mentioned in the reviews follows:

Require: review r_i, general corpus BNC, domain corpus d
Ensure: set of main aspects A and relevance scores
  $A = \{\}$, $T = nouns(r_i)$
  for all $t_k \in T$ **do**
    if $\delta_{t_k}(d||BNC) > \epsilon$ **then**
      $a_{ij} \leftarrow t_k$
      $A = A \bigcup \{a_{ij}\}$
      $rel(a_{ij}, r_i) \leftarrow \delta_{t_k}$
    end if
  end for

[1]Documents and reviews can be used as synonyms, since the framework can be exploited to extract relevant aspects from any kind of textual document.
[2]http://www.natcorp.ox.ac.uk/

Given a review r_i we extract all the nouns from the review and we calculate KL-divergence for each of them. Those having a KL-divergence greater then a threshold ϵ are labeled as *main aspects*, and the KL-divergence is used as relevance score for that aspect.

Next, given the set A containing the main aspects, our framework identifies the sub-aspects related to each $a_{ij} \in A$. To this end, we exploited two measures already used in literature for keyphrase extraction [22], as the *phraseness* and the *informativeness* of a couple of terms. Specifically, the *phraseness* calculates how likely the terms together form a phrase, while the *informativeness* calculates how much is the gain in information when the terms are modeled together. Due to space reasons, we can't provide more details on the extraction of aspects and sub-aspects. We suggest to refer to [3] for a more thorough description of the framework.

However, given such measures, for each main aspect a_{ij} we passed again through all the terms t_k in the review r_i and we labeled as sub-aspect a_{ijk} all the t_k whose *phraseness* and *informativeness* overcome a threshold γ. At the end of the process we obtain a *hierarchy* of aspects and sub-aspects extracted from the review, as that presented in Figure 1. In this case, *soup* is automatically identified as sub-aspect of the main aspect *food*, but food is also a main aspect, so two opinions emerge from the analysis of this review: a good one about the *food* in general, and an higher one specifically related to the *soup*. As previously stated, the process of building such a hierarchy is totally unsupervised, thus we are not bound to a fixed set of aspects, as it happens in most of the current literature, but we dynamically define the *aspects* on the ground of what people actually *think* (and *write*) about a specific item.

Finally, our framework provides all the aspects and sub-aspects extracted from the review with a sentiment score $sent(a_{ijk}, r_i)$. As regards sentiment analysis, we exploited two alternative strategies: a model-based sentiment analysis algorithm included in Stanford CoreNLP[3] which uses deep learning techniques [21], and a lexicon-based algorithm [17] based on the AFINN wordlist [18], which contains about 2500 English words that have been manually tagged with a score that can range from positive (+5) to negative (−5). In both cases, the algorithms return a sentiment score associated to each aspect and sub-aspect extracted from the review, according to the sentiment of the piece of text the aspect is mentioned in.

Multi-Criteria Recommendations. Given such a pipeline, we are able to extract the aspects, the sub-aspects, their relevance (according to KL-divergence) and the sentiment they convey from each specific review. Such a representation is then used to feed a multi-criteria recommendation algorithm based on CF techniques.

Specifically, we considered the sentiment scores extracted by our framework as the *ratings* the user expressed, and we exploited the multi-dimensional euclidean distance proposed by Adomavicius et al. [1] to calculate how similar two *users* (or two *items*) are according to the opinions expressed in the reviews. In the following, we will show how our multi-criteria recommendations are generated in both *user-to-user* and *item-to-item* scenarios.

Multi-criteria User-to-User CF. Similarity between users u_j ed u_k is calculated as the opposite of the distance $dist(u_j, u_k)$, defined as:

[3]http://stanfordnlp.github.io/CoreNLP/

$$dist(u_j, u_k) = \frac{1}{|I(u_j, u_k)|} + \sum_{i \in I(u_j, u_k)} d(R(u_j, i), R(u_k, i)) \quad (2)$$

where $I(u_j, u_k)$ is the set of the co-rated items and $R(u_j, i)$ is the rating expressed by user u_j on item i. Next, if n aspects are mentioned in both reviews, the overall distance is calculated as:

$$d(R(u_j, i), R(u_k, i) = \sqrt{\sum_{a=1}^{n} |R_a(u_j, i) - R_a(u_k, i)|^2} \quad (3)$$

Finally, rating prediction of user u on item i is calculated as the *weighted sum approach* [1] on the top-k neighbors of the user, as:

$$R(u, i) = \sum_{j=1}^{k} \frac{sim(u, u_j) \cdot R(u_j, i)}{|sim(u, u_j)|} \quad (4)$$

Multi-criteria Item-to-Item CF. The similarity between two items i_j and i_k is again the opposite of their distance $dist(i_j, i_k)$ calculated as:

$$dist(i_j, i_k) = \frac{1}{|U(i_j, i_k)|} + \sum_{u \in U(i_j, i_k)} d(R(u, i_j), R(u, i_k)) \quad (5)$$

where $U(i_j, i_k)$ is the set of users who expressed a rating for both i_j and i_k. Again, if n aspects are discussed in the review, then the multi-dimensional distance is calculated as:

$$d(R(u, i_j), R(u, i_k)) = \sqrt{\sum_{c=0}^{n} |R_c(u, i_j) - R_c(u, i_k)|^2} \quad (6)$$

Finally, ratings $R(u, i)$ are predicted using the *weighted sum approach* as it happened in the user-to-user scenario.

$$R(u, i) = \frac{\sum sim(i, j) \cdot R(u, j)}{\sum_{u \in U(i,j)} |sim(i, j)|} \quad (7)$$

In the experimental session we evaluated the effectiveness of both multi-criteria algorithms on varying of different datasets as well as of different configurations of our framework for aspect extraction and sentiment analysis.

3 EXPERIMENTAL EVALUATION

Experiments were carried out on the ground of the following research questions: **(1)** Which combination of the parameters of our framework for aspect extraction and sentiment analysis leads to the best predictive accuracy of our multi-criteria RS? **(2)** How does our algorithm perform, when compared to a single-criteria CF algorithm as well as to other state-of-the-art CF techniques based on matrix factorization?

Experimental design. Experiments were carried out against three state-of-the-art datasets used for evaluating the effectiveness of review-based RS, as *Yelp*[4], *TripAdvisor*[5] and *Amazon*[6]. Table 1 reports some statistics about the datasets.

Our framework for aspect extraction and sentiment analysis was run by comparing different combinations of parameters: first, reviews were processed by removing stop-words and by identifying entities and collocations. Next, the number of aspects/sub-aspects extracted by the framework was set to 10 and 50. In order to assess about the effectiveness of *sub-aspects*, in some experimental session only the main aspects were extracted. As sentiment analysis algorithm, both CoreNLP and AFINN-based algorithms were exploited. The value ϵ, used to define threshold for KL-divergence score, was set to 0.1. As multi-criteria recommendation algorithm, we used both *user-based* and *item-based* CF. The previously introduced multi-dimensional Euclidean distance was exploited to calculate the neighborhood. Neighborhood size was set to 10, 30 and 80 for all the datasets. We used these values since bigger neighborhoods lead to a decrease in the performance of the algorithm.

Table 1: Datasets Statistics

	Yelp	TripAdvisor	Amazon
Users	45,981	536,952	826,773
Items	11,537	3,945	50,210
Ratings/Reviews	229,906	796,958	1,324,759
Sparsity	99,95%	99,96%	99,99%

As regards the baselines, we used single-criteria *user-based* and *item-based CF* [12]. In this case, Euclidean distance and Pearson correlation was used to calculate similarities and neighborhood was set again to 10, 30 and 80. Finally, we also evaluated our algorithm against matrix factorization techniques, as the implementations of Stochastic Gradient Descent (SGD), ParallelSGD [19] and ALSWR [23] available in Mahout[7]. For each baseline we will only report the results of the best-performing combinations of parameters.

The whole experimental session was carried out as a 10-fold cross-validation. The effectiveness of our algorithms was calculated by averaging the Mean Average Error (MAE) obtained on each fold. In order to ensure the reproducibility of the results, metrics were calculated by using Rival [20] framework.

Discussion of the results. In this first experiment we evaluated different configurations of our framework for *aspect-based sentiment analysis*. Results are shown in Table 2 and 3. Due to space reasons, we only report the results obtained with AFINN sentiment analysis algorithm, since it did not emerge any significant difference with those obtained with CoreNLP algorithm. As regards *multi-criteria user-based CF*, on both *Yelp* and *Tripadvisor* the best results are obtained by extracting the *top-10 aspects* gathered from the data. Moreover, on both datasets and for all the neighborhood sizes, the results obtained with 10 aspects are better than those obtained with 50 aspects. Due to this reason, we did not evaluated a larger space of aspects. Another interesting outcome emerging on *Yelp* and *TripAdvisor* is that the use of sub-aspects further improve the performance of the framework, since in all the comparisons the extraction of sub-aspects leads to a lower MAE. However, such results are not confirmed on Amazon data. In this case, the best results are obtained by using top-50 aspects (without sub-aspects). Further investigations are needed to better understand this behavior. It is worth to note that the best-performing configurations for all datasets is that exploiting the top-10 neighbors on all the datasets.

[4]https://www.kaggle.com/c/yelp-recruiting/data
[5]http://www.cs.cmu.edu/ jiweil/html/hotel-review.html
[6]http://jmcauley.ucsd.edu/data/amazon/links.html
[7]https://mahout.apache.org/

Table 2: Results of Experiment 1 for *multi-criteria user-based CF*. The best-performing configuration is highlighted in bold.

Configuration			Dataset		
#neigh.	#asp.	sub-asp	Yelp	TripAdvisor	Amazon
10	10	Y	**0.8362**	**0.7111**	0.6464
10	10	N	0.8410	0.7564	0.6335
10	50	Y	0.8410	0.7269	0.6346
10	50	N	0.8364	0.8007	**0.6276**
30	10	Y	0.8461	0.7677	0.7120
30	10	N	0.8473	0.7722	0.7122
30	50	Y	0.8474	0.7743	0.7101
30	50	N	0.8494	0.8003	0.7140
80	10	Y	0.8579	0.7971	0.7584
80	10	N	0.8592	0.7953	0.7554
80	50	Y	0.8590	0.7907	0.7544
80	50	N	0.8597	0.7995	0.7554

Table 3: Results of Experiment 1 for *multi-criteria item-based CF*. The best-performing configuration is highlighted in bold.

Configuration		Dataset		
#asp.	sub-asp	Yelp	TripAdvisor	Amazon
10	Y	**0.8640**	**0.8245**	**0.8110**
10	N	0.8643	0.8252	0.8117
50	Y	0.8641	0.8254	0.8118
50	N	0.8648	0.8260	0.8124

Table 4: Top-10 *main* aspects extracted for each dataset.

Yelp	place, food, service, restaurant, price menu, staff, drink, lunch, table
Tripadvisor	hotel, room, staff, location, service breakfast, restaurant, bathroom, price, view
Amazon	game, graphic, story, character, player price, gameplay, controller, level, music

Simlilar outcomes are confirmed by analyzing the results obtained by *multi-criteria item-based CF*, even if with a smaller gap. For the sake of completeness, in Table 4 we report the top-10 main aspects extracted from each dataset. Due to space reasons, it is not possible to report more aspects or sub-aspects of each main aspect.

Next, in Experiment 2 we compared the best-performing configuration to several baselines. First, we evaluated the effectiveness of single-criteria CF algorithms, as User-to-User CF (U2U) and Item-to-Item (I2I). Results are reported in Table 5. In this case, we report the values of the two similarity measures, euclidean and Pearson, obtained with the optimal number of neighbors. As we expected, our multi-criteria CF algorithm overcomes all the baselines on all the dataset. Such improvements are obtained on both U2U and I2I algorithm.

Table 5: Results of Experiment 2. The best-performing configuration is highlighted in bold. Next to each baseline we report the similarity measure used in that specific run. N.A. refers to not available results.

	Dataset		
Configuration	**Yelp**	**Tripadvisor**	**Amazon**
Multi-U2U	**0.8362**	**0.7111**	**0.6276**
U2U-Euclidean	0.8860	0.8337	0.7254
U2U-Pearson	0.9640	1.1222	0.9789
Static-Multi-U2U	N.A.	0.7980	N.A.
Multi-I2I	0.8640	0.8245	0.8110
I2I-Euclidean	0.8745	0.8429	0.8177
I2I-Pearson	1.1794	0.8644	0.9679
Static-Multi-I2I	N.A.	0.8474	N.A
RatingSGD	0.8409	0.7450	0.8859
ParallelSGD	0.8409	0.7449	0.8852
ALSWR	0.9545	0.9053	1.0354

Next, we compared our algorithm to more sophisticated baselines based on matrix factorization algorithms. Moreover, given that TripAdvisor dataset has a static set of six aspects for each review *(cleanliness, location, value, service, sleep quality* and *overall*), we also compared our approach to a (*statically defined*) multi-criteria algorithm based on such aspects. As shown in Table 5, our approach overcomes again all the baselines we took into account, those based on matrix factorization and that exploiting a static multi-criteria algorithm. These results definitely confirmed our insights, since we showed that our strategy can overcome both strategies exploiting single ratings (with or without matrix factorization) as well as multi-criteria algorithms that exploit a fixed set of pre-defined aspects. The best overall results are obtained when **multi-criteria User-to-User CF** is used as recommendation algorithm.

4 CONCLUSIONS AND FUTURE WORK

In this paper we presented a multi-criteria RS which exploits users reviews to build a multi-faceted representation of users interests. The novelty of this work lies in the fact that our framework is able to unsupervisedly extract relevant aspects from the review, and it is not bound to a fixed and static set of pre-defined aspects. Moreover, our algorithm also extracts sub-aspects related to the main aspect, thus building a finer-grained representation of the content conveyed by the review. As shown in the experiments, our framework overcame all the baselines taken into account, even those based on sophisticated techniques for matrix factorization.

As future work, we will encode the features extracted from users' reviews on context-aware recommendation algorithms [16]. Moreover, we will also evaluate the use of ontologies and rules [11], in order to implement reasoning mechanisms to better identify the most relevant aspects in the reviews. Finally, we will evaluate how our framework impacts on different evaluation metrics, as the *serendipity* of the recommendations [7]

REFERENCES

[1] Gediminas Adomavicius and YoungOk Kwon. 2007. New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems* 22, 3 (2007).

[2] Gediminas Adomavicius and YoungOk Kwon. 2015. Multi-criteria recommender systems. In *Recommender systems handbook*. Springer, 847–880.

[3] Annalina Caputo, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Gaetano Rossiello. 2017. SABRE: A Sentiment Aspect-Based Retrieval Engine. In *Information Filtering and Retrieval*. Springer, 63–78.

[4] Guanliang Chen and Li Chen. 2015. Augmenting service recommender systems by incorporating contextual opinions from user reviews. *User Modeling and User-Adapted Interaction* 25, 3 (2015), 295–329.

[5] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction* 25, 2 (2015), 99–154.

[6] Marco De Gemmis, Leo Iaquinta, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. 2010. Learning preference models in recommender systems. In *Preference Learning*. Springer Berlin Heidelberg, 387–407.

[7] Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Cataldo Musto. 2015. An investigation on the serendipity problem in recommender systems. *Information Processing & Management* 51, 5 (2015), 695–717.

[8] Yanjie Fu, Bin Liu, Yong Ge, Zijun Yao, and Hui Xiong. 2014. User preference learning with multiple information fusion for restaurant recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 470–478.

[9] Matthias Fuchs and Markus Zanker. 2012. Multi-criteria ratings for recommender systems: an empirical analysis in the tourism domain. *E-commerce and web technologies* (2012), 100–111.

[10] Dietmar Jannach, Zeynep Karakaya, and Fatih Gedikli. 2012. Accuracy improvements for multi-criteria recommender systems. In *Proceedings of the 13th ACM conference on electronic commerce*. ACM, 674–689.

[11] Loredana Laera, Valentina A. M. Tamma, Trevor J. M. Bench-Capon, and Giovanni Semeraro. 2004. SweetProlog: A System to Integrate Ontologies and Rules. In *Rules and Rule Markup Languages for the Semantic Web: Third International Workshop, RuleML 2004, Hiroshima, Japan, November 8, 2004. Proceedings (Lecture Notes in Computer Science)*, Grigoris Antoniou and Harold Boley (Eds.), Vol. 3323. Springer, 188–193.

[12] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[13] Hongyan Liu, Jun He, Tingting Wang, Wenting Song, and Xiaoyang Du. 2013. Combining user preferences and user opinions for accurate recommendation. *Electronic Commerce Research and Applications* 12, 1 (2013), 14–23.

[14] Liwei Liu, Nikolay Mehandjiev, and Dong-Ling Xu. 2011. Multi-criteria service recommendation based on user criteria preferences. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 77–84.

[15] Nikos Manouselis and Constantina Costopoulou. 2007. Experimental analysis of design choices in multiattribute utility collaborative filtering. *International Journal of Pattern Recognition and Artificial Intelligence* 21, 02 (2007), 311–331.

[16] Cataldo Musto, Giovanni Semeraro, Pasquale Lops, and Marco de Gemmis. 2014. Combining distributional semantics and entity linking for context-aware content-based recommendation. In *User Modeling, Adaptation, and Personalization*. Springer, 381–392.

[17] Cataldo Musto, Giovanni Semeraro, and Marco Polignano. 2014. A comparison of lexicon-based approaches for sentiment analysis of microblog posts. *DART 2014: Information Filtering and Retrieval. CEUR-WS.org, Volume 1314* (2014), 59–68.

[18] Finn Årup Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903* (2011).

[19] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*. 693–701.

[20] Alan Said and Alejandro Bellogín. 2014. Rival: a toolkit to foster reproducibility in recommender system evaluation. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 371–372.

[21] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, Vol. 1631. Citeseer, 1642.

[22] Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18*. Association for Computational Linguistics, 33–40.

[23] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. 2008. Large-scale parallel collaborative filtering for the Netflix prize. In *International Conference on Algorithmic Applications in Management*. Springer, 337–348.