An OWA and Aspect-based approach applied to Rating Prediction

Jesus Serrano-Guerrero
Department of Information Technologies and Systems
University of Castilla-La Mancha
Ciudad Real, Spain
jesus.serrano@uclm.es

Francisco P. Romero
Department of Information Technologies and Systems
University of Castilla-La Mancha
Ciudad Real, Spain
franciscop.romero@uclm.es

Jose A. Olivas
Department of Information Technologies and Systems
University of Castilla-La Mancha
Ciudad Real, Spain
joseangel.olivas@uclm.es

***Abstract*—We have witnessed a flourish of review websites where users can buy many products/services and share their opinions about them. Most of those opinions may be broken down into different sub-opinions on the different aspects describing said products/services. This fact makes more complicated the task of computing the overall polarity about the product/service studied.**

We are presenting a fuzzy aggregation mechanism to compute the overall sentiment conveyed in a opinion/review taking into account the individual ratings for the different aspects commented by the opinion holder. This proposal has been tested using real data from Yelp dataset obtaining promising results.

***Index Terms*—Sentiment analysis, Aspect rating, Fuzzy aggregation**

I. Introduction

In our daily life, consumers make decisions about what products to purchase depending on highly-praised opinions/reviews conveyed by other past consumers. These reviews are considered, in many cases, more trustworthy than even the vendor product descriptions. Therefore, customers are highly concerned about product's reputation, which is expressed through these reviews and reflects consumers' evaluation based on the rating of a product or service.

Nevertheless, it is not only necessary to know whether a specific product is good, but also how good said product is. This fact is manually feasible when the number of reviews is small, however, when is high the implementation of an automatic process is necessary, especially, if there are many aspects to be considered about the products to purchase. For example, when booking a hotel, a client might pay attention to different factors such as the quality of food served by the hotel, the location, the additional services offered (wifi, gym, ...), etc.

Aspect-based Sentiment Analysis is a subarea mainly focused on dealing with tasks such as detecting the main features characterizing a product, which might be opinionated by a user and, rating them as well as the product as a whole. In theory, the overall rating should be a composition of the individual rating of the features treated in the opinion.

Some online services like Tripadvisor guide the user to express/rate their opinions on the basis of well-defined features (sleep quality, room, service, etc.) as can be seen in Fig. 1.

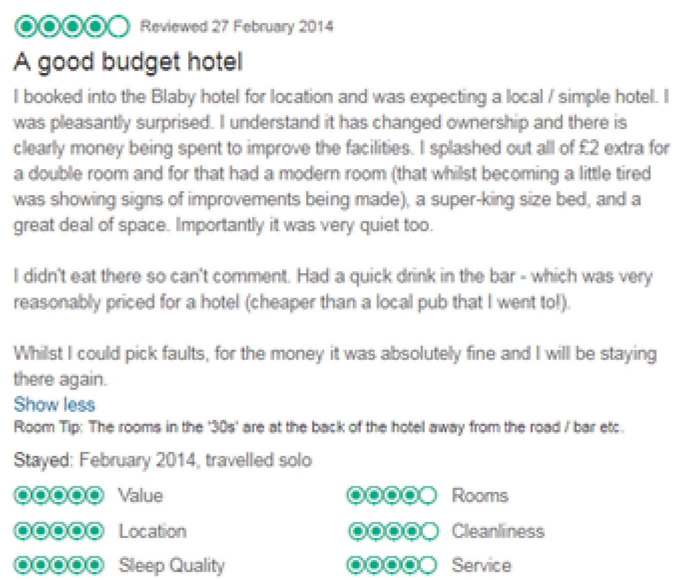

Fig. 1. Architecture of product review aggregation

This fact helps the user understand the review; nevertheless, many other online services allow the user to convey their opinions as free text without any guidelines. In these cases, it is necessary to automatically compute the aspects. In spite of the fact that there are many attempts dealing with this task, it is observed that "only a small number of the detected aspects correspond to ratable aspects and they are not automatically separable" [1]. In our case, we are considering a problem in which, the ratable aspects are clearly defined. This way, it may

This work has been partially supported by FEDER and the State Research Agency (AEI) of the Spanish Ministry of Economy and Competition under grant TIN2016-76843-C4-2-R (AEI/FEDER, UE)

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

be easier to apply a semantic approach to automatically detect the aspects expressed.

Therefore, the challenge here is to rate them and aggregate them in such a way that the overall rating is calculated as composition of the individual aspect ratings. In this sense, Fuzzy Logic provides us with a wide range of aggregation operators [2] which allow us to model in a flexible way the different variables of the problem.

Moreover, it is necessary to keep in mind different factors like, for example, that a review, not necessarily must contain information about all aspects, or the user might consider that some aspects are more important than others when expressing the overall rating. Therefore, based on all these facts, this paper tries to cope with the problem of rating detection proposing as main contributions:

- an algorithm for aggregating individual aspect ratings to compute the overall rating of an opinion,
- a possible implementation of these ideas and,
- a case study, analyzing the results obtained taking into account different parameters.

The remainder of the paper is organized as follows: Section 2 describes some works related to this one and Section 3 presents some mathematical definitions necessary for our proposal presented in Section 4. Section 5 points out a possible implementation along with some experimental results, and finally, Section 6 describes some conclusions reached as well as some future work.

II. State of the Art

A. Sentiment Analysis

Sentiment Opinion, also called Opinion Mining, is mainly conducted on three main different levels: document-level, sentence-level, and aspect-level [3], [4]. This work is mainly focused on issues related to the aspect-based level.

Usually, a typical aspect-based sentiment analysis system implies two phases. First of all, it extracts aspects and then, it computes the sentiment of the said aspects.

The concept of an aspect may be slightly different depending on the work, especially, when they are detected automatically. An aspect might be modelled as a frequent noun or noun phrase [5], or a cluster o a set of words [1], [6].

There are two possible approaches to detect aspects: domain-dependent or domain-independent. The latter does not require predefined aspects or a domain-dependent sentiment lexicon [6]–[9], however, the quality of the results is poorer than the domain-dependent approach, because most of times the polarity of a term depends on the domain [10].

To compute the aspect ratings, there are strategies based on the use of semantic resources like ontologies or thesaurus such as HowNet or SentiWordNet [6]–[9].

Nevertheless, there are so many different domains to model and define the corresponding aspects that it is almost impossible to have an associated lexicon/thesaurus for all of them. Hence, it is vital to work on alternatives like non-supervised domain-independent models for aspect-based sentiment analysis.

For that reason, many new works based on Latent Dirichlet Allocation (LDA) have been appeared recently [11]. For example, Lei et al. [12] proposed a mechanism for extracting features using LDA. In this case, different topics are found, represented by the most descriptive words. Using Yelp [1], the restaurant opinions dataset, the topics seem to represent the main features for this domain, for example: location, price, etc.

Also based on LDA, the Joint Sentiment Topic Model presents a slight difference [13], instead of modelling topics or sentiments separately, it models topics and sentiments jointly as the Aspect and Sentiment Unification Model does, but considering that all words from any sentence come from the same language model [14]. Nevertheless, these models do not perform well when there are many uncorrelated topics. In this case, other options like Multi-grain Topic Models go a step further, outperforming the previous models [1].

New trends based on Deep Learning are also arising, as a complement to LDA. In this sense for example, Ma et al. improved aspect detection by using LDA and including the use of Word2vec [7], but many other works can be found proposing just based-Deep Learning solutions [15], [16].

Moreover, another challenge to face is to know the optimal number of aspects required to model online reviews. Sometimes, the number is very small or high when they are calculated automatically [17], and a manual adjustment step is necessary [1].

Apart from detecting aspects and their associated polarity, other works especially-focused on personalization, take into account the importance of each aspect. Different strategies to calculate these weight have been developed. The user may express explicitly the importance [18] or the system can automatically estimate them through the information available about the opinions [6]. In the latter case, the importance weights depend on three factors: the user, item and aspect and, it is necessary to compute the rating for individual aspects and know the overall rating given by the user to the product, to estimate the importance of each aspect.

Once all previous elements (features, weights, sentiments) have been computed, the aggregation process can be computed at once as explained in previous works [1] or alternatively, it can be carried out as shown in Fig. 2, used in approaches like [18], [19]:

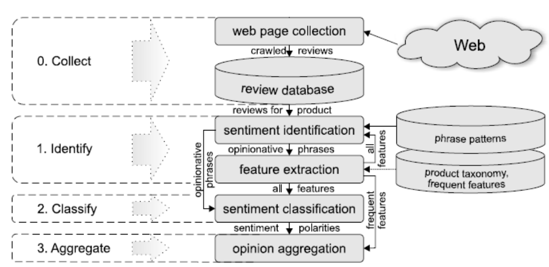

Fig. 2. Architecture of product review aggregation

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

B. Ordered Weighted Averaging (OWA) operator

An Ordered Weighted Averaging (OWA) operator of dimension n is defined as a mapping $F : R^n \rightarrow R$ which has an associated weighting vector $W = [w_1, w_2, \ldots, w_n]$ in which $w_j \in [0,1]$ and $\sum_{j=1}^{n} w_j = 1$ and where $F(a_1, a_2, \ldots, a_n,) = \sum_{j=1}^{n} w_j b_i$, with b_j being the j-th largest of the collection of the aggregated objects a_i.

One of the main aspects of the OWA operators is a reordering step, each element a_i is not associated with a particular weight w_i but each weight w_i is associated with a particular position i of the ordered elements. The type of aggregation performed by an OWA operator depends upon the form of the weighting vector, therefore, by the selection of the appropriate vector W, the OWA operators can model the max, min and arithmetic mean operators. Thus, given the vectors $W = [1, 0, 0, \ldots, 0]$, $W = [0, 0, 0, \ldots, 1]$ and $W = [1/n, 1/n, \ldots, 1/n]$, then $f(a_1, \ldots, a_n) = Max(a_i)$, $f(a_1, \ldots, a_n) = Min(a_i)$ and $f(a_1, \ldots, a_n) = \frac{1}{n} \sum_{i=1}^{n} a_i$, respectively.

Two characterizing measures associated with the weighting vector W of an OWA operator were introduced by Yager [20]. The first one is known as the measure of orness of the aggregation and is defined as

$$orness(W) = \frac{1}{n-1} \sum_{i=1}^{n} (n-i) w_i \tag{1}$$

This measure characterizes the degree to which the aggregation is similar to an OR operation. On the other hand the other measure calculates the dispersion of the aggregation and is defined as

$$disp(W) = -\sum_{i=1}^{n} w_i \ln w_i. \tag{2}$$

On the contrary, this measure assesses the degree to which W takes into account all information in the aggregation.

The problem of choosing the weights for an OWA operator can be addressed in different ways, for example, by the so-called linguistic quantifiers, introduced by L. Zadeh in [21].

Linguistic quantifiers Q such as most (see Fig. 3), few, many and all, can be represented as a fuzzy subset of the unit interval, $Q(r)$ indicates the extent to which a given proportion $r \in [0,1]$ of the total of values to aggregate, satisfies the semantics defined in Q.

Yager proposed the Regularly Increasing Monotonic (RIM) quantifiers as a way of obtaining a weighting vector W associated with an OWA aggregation [22]. These quantifiers present the following properties:

- $Q(0) = 0$
- $Q(1) = 1$
- If $r1 > r2$ then $Q(r1) \geq Q(r2)$.

According to Yager [20], given a RIM quantifier Q, we can compute an OWA weighting vector W associated with Q such that for $j \in \{1, \ldots, n\}$:

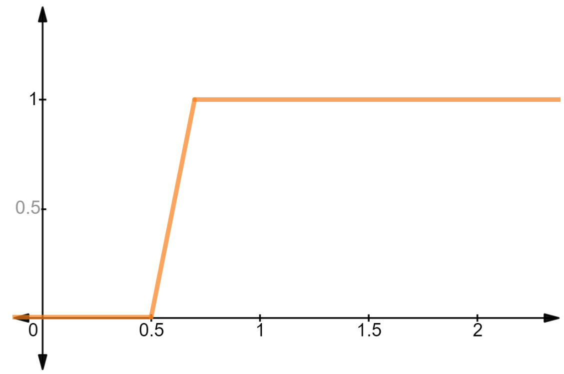

Fig. 3. Example of Q="most" linguistic quantifier

$$w_j = Q\left(\frac{j}{n}\right) - Q\left(\frac{j-1}{n}\right) \tag{3}$$

where the membership function of a linear RIM quantifier Q(r) is defined by two parameters $a, b \in [0,1]$ as

$$Q(r) = \begin{cases} 0 & \text{if } r < a \\ \frac{r-a}{b-a} & \text{if } a \leq r \leq b \\ 1 & \text{if } r > b \end{cases} \tag{4}$$

Therefore, the linguistic expression of the quantifier can help us calculate the weighting vector W and to include any meaning in the aggregation.

III. Definitions

To understand the proposal, it is necessary to define the mathematical elements taking part, which are defined in this section.

Given a set of opinions/reviews $D = \{d_1, d_2, \cdots, d_{|D|}\}$ about a product or topic, each opinion/review $d \in D$ has an overall rating r_d and there are also n unique terms from the vocabulary $V = \{t_1, t_2, \cdots, t_n\}$ modelled by the opinions/reviews.

A. Overall Rating

An overall rating r_d of a review/opinion d is a numerical value indicating the polarity expressed by d. $r_d \in [r_{min}, r_{max}]$, where r_{min} and r_{max} are the minimum and maximum values, respectively. The higher the value r_d the more positive the polarity is.

The topic treated in d can express ideas about k aspects, which may be considered as potential subfactors affecting the overall rating.

B. Aspect

An aspect A_i is typically a noun, noun phrase or even a set of words characterizing a ratable factor in the reviews/opinions. For example, for a restaurant review, "price" or "style" can be aspects characterizing said restaurant.

C. Aspect Ratings

The aspect ratings S are represented by a k dimensional vector, where the $i-th$ position is a numerical value ($s_{di} \in [r_{min}, r_{max}]$), indicating the polarity in the review d towards the aspect A_i.

D. Aspect Weights

The aspect weights W are represented by a k dimensional vector, where the $i-th$ position is a numerical value w_{di}, modelling the importance degree of the aspect A_i treated in the review d. The higher the weight is, the more important it is. In this case, $w_{di} \in [0,1]$ and $\sum_{i=1}^{k} w_{di} = 1$.

E. Aspect Rating Aggregator

The aspect rating aggregator $agg(S,W)$ is a mathematical operator which combines the vectors Aspect Weights W and Aspect Ratings S in order to compute the overall rating $r_d \in [r_{min}, r_{max}]$ of the review d.

IV. Proposal

This paper presents a mechanism for computing the overall rating of an opinion through the aggregation of the individual ratings for each aspect detected. A summary of the proposal has been depicted in Figure 4.

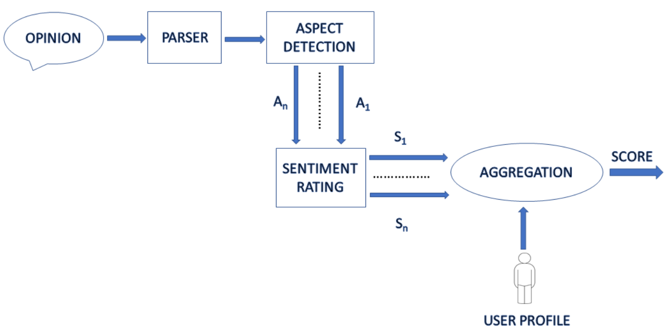

Fig. 4. Steps of the algorithm

The main steps of the proposed mechanism are:

- Pre-processing step:
  The system input consists of a plain text representing the opinion conveyed. A parser must pre-process the whole text, focusing on the main task of this phase, which is the sentence splitter. As we are working at sentence level, it is necessary to implement an algorithm for detecting the different sentences contained.
  Aside from the sentence splitter, this parser may implement other typical Natural Language Processing tasks like removing stopwords, lemmatizing, etc., depending on the type of text we are working with.
- Aspect detection:
  After finding the sentences from the opinion, it is necessary to detect the main aspect each sentence is talking about. As it was commented in the state of art, two possible strategies can be found. If the aspects to be considered are previously known, a classification algorithm must be used to correctly classified each sentence. If not, it is necessary to previously detect the main aspects treated for all opinions, some examples have been mentioned in the state of the art, for example, using techniques like LDA. Once these main aspects have been automatically detected, then each sentence from the previous step must be classified in the corresponding aspect A_i.
- Aspect rating:
  After detecting and classifying each sentence, although this step might be concurrently computed to the previous one, it is necessary to rate each sentence. In this case, it is not only necessary to categorize the opinion as positive or negative, but it is also necessary to compute a score s_{di} grading to what extent the opinion is positive or negative.
- Aggregation:
  As it was presented in the state of the art, the use of the OWA operator provides us with some flexibility to aggregate different scores.
  In this case, the importance of the different aspects W will depend on the user character. For example, an optimistic character would consider the positive aspects to contribute more to final score than the negatives. Nevertheless, a negative character would consider the negative aspects to contribute in a more critical way to the final score. That would allow modelling opinions like the one seen in Figure 5 from [18], where the most negative aspects weigh more than the positive ones when calculating the overall rating. Therefore, depending on the orness of the operator used, different user behaviours may be implemented.
  For example, having 5 aspects, if W were $W = [0, 0, 0, 0.2, 0.8]$, then the most positive aspects, due to the reordering step, would be aggregated; in this case, the operator would be closer to calculate the maximum. On the other hand, if W were $W = [0.8, 0.2, 0, 0, 0]$, then the most negative values would be aggregated; in this case, this operator would be closer to calculate the minimum.
  Hence, different OWA operators can be proposed as aspect rating aggregators for this proposal.

V. Experiments

A. Dataset

The dataset used is provided by Yelp website for the Yelp Dataset Challenge [2] and it mainly contains reviews and ratings given by Yelp website's users about business activities, mainly restaurants.

The Yelp dataset 2018 includes several files in JSON format such as User (i.e., Yelp's registered members), Business, Review written by a User on a specific Business, Tip given by

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

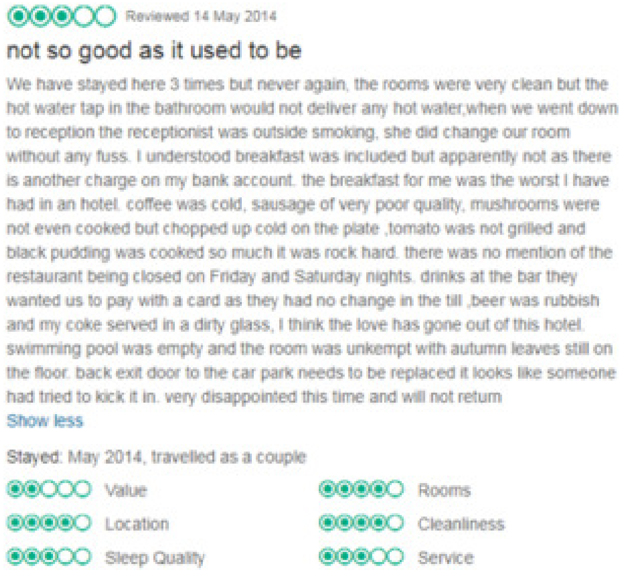

Fig. 5. Example of opinion contributing more the most negative aspects to final score

User, among others. In this work, we are especially interested in the Reviews.

Each record representing a Review includes fields like review_id, user_id (the user who wrote the review), business_id (the business opinionated), stars (score using a scale from $[1, 5]$ rating the business), date, useful, funny, cool and text. The Review fields like useful, funny or cool are integer values to indicate usefulness and sentiment of the review, but not always are provided.

B. Implementation

First of all, it is necessary to decide how to compute the different aspects to be treated in this problem. Yelp data collection only provides the overall rating for an opinion, whereas others like Tripadvisor allow the user to score a set of predefined individual aspects as can been seen in Figure 5.

Although Yelp dataset does not provide aspects, the restaurant domain is very well-known, for that reason, it is easy to define a set of aspects which will be opinionated in most of the reviews. In this case, the use of a semantic strategy based, for example, on the use of an ontology on that domain to characterize all concepts belonging to each aspect, seems quite appropriate.

Several strategies can be found in the state of the art, but as the main point of this proposal is the aggregation step, and in order to simplify the process, we have decided to use a REST service provided by Aylien[3] which is based on domain taxonomies. In this case, a specific service for restaurants is provided, which divides opinions into different sentences classified under the following aspects: ”food”, ”staff”, ”ambience”, ”menu”, ”location”, ”reservations”, ”cleanliness”, ”desserts”, ”drinks”, ”value”, ”payment”, ”business”, ”quietness”, ”facilities” and ”entertainment”.

This REST service also implements an algorithm for computing the sentiment rating for each sentence, which makes this tool especially suitable for focusing on the aggregation step.

Once the sentences have been detected as well as the corresponding aspects and ratings, the OWA-based aggregation has been implemented by the Java library provided by Torra [4] in order to compute the final rating.

C. Evaluation measures

The experiments are evaluated by using the Mean Absolute Error (MAE) measure, which calculates the absolute difference between the actual rating (R_a) and the predicted rating (R_p). The lower the value of MAE is, the closer the predicted rating to the actual rating is. Let R be a set of reviews r:

$$MAE = \frac{1}{|R|} \sum_{r \in R} |R_a - R_p| \quad (5)$$

D. Results

For these experiments, the 50,000 first opinions from Yelp have been processed by Aylien and analyzed from two different points of view.

Considering the attitude of the users, different types of aggregations may be planned. The more positive a user is, the more he is going to take into account the positive aspects in a review when giving the overall rating. Nevertheless, the more negative a user is, the more he is going to taking into account the negative aspects in a review when giving the overall rating. As it was explained before, this fact may be modelled through different combination of weights W of the OWA operator, i.e., those OWA operators which are closer to represent an OR operation may be represent a positive user and, those which are close to AND operation, may represent a negative user. To check this, different RIM quantifier with different orness (see equation (1)) have been proposed:

- RIM 0.7-1 (Orn 0.15): The RIM quantifier has been computed taking $a = 0.7$ and $b = 1$ (see equation 4). Its orness is 0.15.
- RIM 0.75-0.9 (Orn 0.2): The RIM quantifier has been computed taking $a = 0.7$ and $b = 0.9$. Its orness is 0.2.
- RIM 0.7-0.8 (Orn 0.25): The RIM quantifier has been computed taking $a = 0.7$ and $b = 0.8$. Its orness is 0.25.
- RIM 0.45-0.75 (Orn 0.4): The RIM quantifier has been computed taking $a = 0.45$ and $b = 0.75$. Its orness is 0.4.
- RIM 0.45-65 (Orn 0.45): The RIM quantifier has been computed taking $a = 0.45$ and $b = 0.65$. Its orness is 0.45
- RIM 0.45-0.55 (Orn 0.5): The RIM quantifier has been computed taking $a = 0.45$ and $b = 0.55$. Its orness is 0.5.
- RIM 0.5-0.65 (Orn 0.65): The RIM quantifier has been computed taking $a = 0.2$ and $b = 0.5$. Its orness is 0.65.

[3]https://aylien.com

[4]http://www.mdai.cat/ifao/wowa.php

- RIM 0.2-0.4 (Orn 0.7): The RIM quantifier has been computed taking $a = 0.2$ and $b = 0.4$. Its orness is 0.7.
- RIM 0.2-0.3 (Orn 0.75): The RIM quantifier has been computed taking $a = 0.2$ and $b = 0.3$. Its orness is 0.75.

According to those possible combinations, the MAEs obtained applying this approach can be seen in Figure 6.

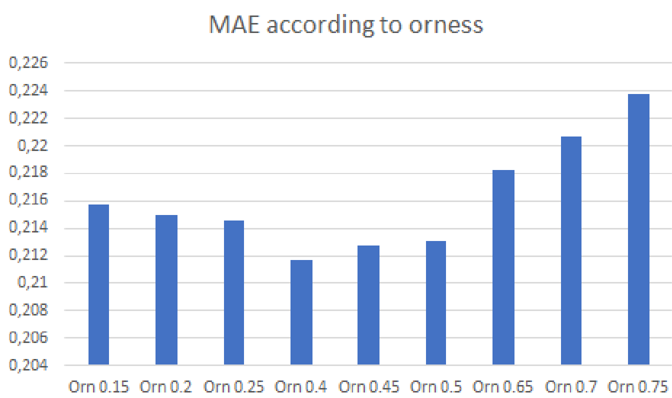

Fig. 6. MAE depending on the orness

From these results, it is easy to see that the extremes are less valuable than the central values. This fact may be due to fact that most users do not tend to be radical, just balanced and fair when expressing their opinions.

This might be a general idea, but these data may be broken down from different perspectives. Firstly, taking into account the number of aspects detected by the system and secondly, considering the number of stars per review. Thus, all reviews qualified by 1 star have been processed separately, all qualified by 2 stars separately and so on; and the same process has been followed grouping all reviews containing just 2 aspects separately, just 3 aspects separately, and so on.

Thus, following that idea and considering the number of stars per review, the errors calculated are depicted in Fig. 7.

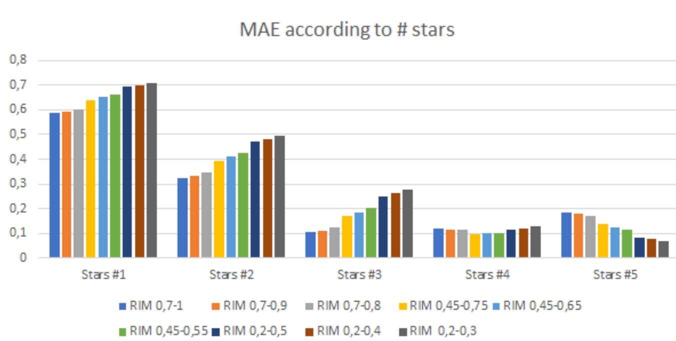

Fig. 7. MAE depending on the number of stars

In this case, the proposal works better as the number of stars is greater. It may be thought that there were a little number of low-rated reviews, however, almost 12% of the reviews analyzed have 1 star-rating, which seems a reasonable number of reviews to test. Around 8% were 2-star reviews, 13% 3-star reviews, 29% 4-star reviews and 38% 5-star reviews.

TABLE I
MAE PER ASPECTS #1, #2, #3 AND #4

	Aspe. #1	Aspe. #2	Aspe. #3	Aspe. #4
RIM 0.7-1	0.2259226	0.2292632	0.2260012	0.2208984
RIM 0.7-0.9	0.2259226	0.2292632	0.2260012	0.2193153
RIM 0.7-0.8	0.2259226	0.2292632	0.2260012	0.2158841
RIM 0.45-0.75	0.2259226	0.2238550	0.2166843	0.2169530
RIM 0.45-0.65	0.2259226	0.2216748	0.2200712	0.2170912
RIM 0.45-0.55	0.2259226	0.2172214	0.2200712	0.2189023
RIM 0.2-0.5	0.2259226	0.2195413	0.2211172	0.2300459
RIM 0.2-0.4	0.2259226	0.2195413	0.2247083	0.2309316
RIM 0.2-0.3	0.2259226	0.2195413	0.2317600	0.2348875

Furthermore, Fig. 6 might lead us to think that the fact of having more positive reviews from Yelp (4 and 5-star reviews) might affect those quantifiers with greater orness, which are obtaining worse results than the ones with lower orness as can be seen in Fig. 6. Nevertheless, the reason is mainly that they make greater errors when the number of stars is 3 or less as it can be seen in Fig. 7.

Analyzing some reviews, low-rated reviews are mostly low informative, the descriptions offered are not very descriptive, using adjectives which are not very representative for very negative opinions, like ”plain” instead of ”awful” or ”terrible”. To exemplify this, a 1-star review which does not seem very descriptive may be:

”I too have been trying to book an appt to use my voucher - it's been months and countless of phone calls - no response yet.Agree with the buyers beware warning. I only wish reviews on this place was posted previous to my purchase of this voucher.”

That might explain why the error computed in this case, is always greater when the number of stars is low, in contrast to the highly-rated opinions.

From Fig. 7, it is also possible to get some information paying attention to the orness of the quantifiers modelling the different user behaviours.

The low-rated opinions, 1 star for example, obtain less MAE when using low-orness quantifiers, whereas the highly-rated opinons, 5 stars for example, obtain less MAE when using high-orness quantifiers. This may be explainable because if the users had a good stay in a hotel or restaurant, then their opinion would be very positive without analyzing objectively the quality of the services offered, whereas in case of having a bad experience or just some incidents during their stay, they would tend to assess the hotel or restaurant in a negative way, just because they are angry, even when some aspects of the stay may be described as good.

And on the other hand, paying attention to the number of aspects per review, Table 1 and 2 and 3 show the MAE computed.

The maximum number of aspects detected in a review has been 12, in spite of the fact that it is possible to count up to 15 aspects. However, it is very complicated that a user talks about all possible aspects in a review.

TABLE II
MAE PER ASPECTS #5, #6, #7 AND #8

	Aspe. #5	Aspe. #6	Aspe. #7	Aspe. #8
RIM 0.7-1	0.2205730	0.2187893	0.2137856	0.2123060
RIM 0.7-0.9	0.2188606	0.2175129	0.2104099	0.2087614
RIM 0.7-0.8	0.2201534	0.2187714	**0.2085804**	**0.2077209**
RIM 0.45-0.75	0.2206189	0.2216891	**0.2066458**	**0.2070601**
RIM 0.45-0.65	0.2234255	0.2274318	**0.2098251**	**0.2097108**
RIM 0.45-0.55	0.2278140	0.2309714	0.2163253	0.2141789
RIM 0.2-0.5	0.2352548	0.2487027	0.2295775	0.2332962
RIM 0.2-0.4	0.2426736	0.2535367	0.2372419	0.2409335
RIM 0.2-0.3	0.2426736	0.2616378	0.2449680	0.2480870

TABLE III
MAE PER ASPECTS #9, #10, #11 AND #12

	Aspe. #9	Aspe. #10	Aspe. #11	Aspe. #12
RIM 0.7-1	**0.2097750**	0.2261831	0.2160633	0.2404758
RIM 0.7-0.9	**0.2056544**	0.2207292	0.2101673	0.2574749
RIM 0.7-0.8	**0.1998887**	0.2181876	0.2194598	0.2664757
RIM 0.45-0.75	**0.1962579**	0.2190069	0.2656769	0.2988739
RIM 0.45-0.65	**0.2000842**	0.2240445	0.2796604	0.3088896
RIM 0.45-0.55	**0.2064474**	0.2317167	0.2908924	0.3255821
RIM 0.2-0.5	0.2287173	0.2513546	0.3045837	0.3761646
RIM 0.2-0.4	0.2388516	0.2570194	0.3112829	0.3836446
RIM 0.2-0.3	0.2464424	0.2613500	0.3227023	0.3929712

As it can be seen, the best results have been obtained when there is a substantial but not huge number of aspects. Working with 7, 8 and 9 aspects, the lowest values were obtained (marked in bold). Nevertheless, for most cases the worst results were obtained when there are many or few aspects, i.e., the extremes. Analyzing some examples, when there is no too much information, just a few aspects, the user seems to omit some information or focus on some specific aspects, which may affect the final rating but are not expressed in the opinion. And when the information is overwhelming, the results are worse because it seems complicated to establish exactly which the most determining factors to compute are. As a result, the most appropriate reviews are those which are descriptive but are not containing excessive information.

VI. CONCLUSIONS AND FUTURE WORK

A novel aspect-based rating prediction method based on fuzzy aggregation is proposed, which considers that the individual aspect ratings must compose the overall rating of any review. The model developed is mainly based on the aspect ratings S and the importance of said aspects modelled through the weight vector W of the OWA operator. Depending on those weights, it is possible to represent different user attitudes.

This approach has been evaluated by using the restaurant reviews of Yelp's dataset. The experimental results show that the proposed method can predict accurately the overall rating of any opinion, assessing previously the individual rating of the different aspects. To do so, it is necessary to pay attention to the different user profiles simulated. For example, when the opinion rating is low, it is better to simulate a negative user by using a scheme of weights with low orness. Nevertheless, if the opinion is positive, it is better to use a RIM quantifier with high orness.

It has also been tested that when the number of aspects is substantial but not overwhelming, the proposal performs better. This seems to be due to the fact that this type of opinions are more informative/descriptive than short reviews based on a few aspects and the information which are providing is just the necessary, in contrast to the opinions expressing many aspects.

As future works, it is necessary to pay attention to other details, for example, the implicit information. Not always the overall rating for a product takes into account the information conveyed in an opinion. Sometimes, there are other factors which are not literally written by the opinion holder, possibly only those which the user considers as more relevant for the reader, were expressed. This fact is especially important to aspect-based systems because when there are many aspects to assess, it is not easy to get the reader to rate/comment all of them. Therefore, it might be interesting to detect and/or include some other hidden/implicit factors to adjust the final rating computed.

It is also necessary to see the data from other different points of view, for example, in the case of hotels, the cheapest ones tend to obtain worse opinions [23] than the most expensive hotels because they offer more luxurious services. Therefore, price or services may be another factor to be analyzed.

Moreover, there are many more operators from the OWA family like WOWA, LOWA, IWOWA, etc., which should be studied to model these other mentioned factors from users and opinions. [24]–[26].

REFERENCES

[1] M. Alam, W. Ryu, and S. Lee, "Joint multi-grain topic sentiment: modeling semantic aspects for online reviews," *Information Sciences*, vol. 339, pp. 206–223, 2016.

[2] R. R. Yager, "Families of OWA operators," *Fuzzy Sets and Systems*, vol. 59, no. 2, pp. 125 – 148, 1993.

[3] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, 2014.

[4] J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, and E. Herrera-Viedma, "Sentiment analysis: A review and comparative analysis of web services," *Information Sciences*, vol. 311, no. 1, pp. 18–38, 2015.

[5] Z. Zha, Y. J., J. Tang, M. Wang, and T.-S. Chua, "Product Aspect Ranking and Its Applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1211–1224, 5 2014.

[6] A. Da'u, N. Salim, I. Rabiu, and A. Osman, "Weighted aspect-based opinion mining using deep learning for recommender system," *Expert Systems with Applications*, vol. 140, p. 112871, 2020.

[7] X. Ma, X. Lei, G. Zhao, and X. Qian, "Rating prediction by exploring user's preference and sentiment," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 6425–6444, 2018.

[8] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-Based Methods for Sentiment Analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[9] C. Wu, F. Wu, J. Liu, Y. Huang, and X. Xie, "Sentiment Lexicon Enhanced Neural Sentiment Classification," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management - CIKM '19*. ACM Press, 2019, pp. 1091–1100.

[10] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[11] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[12] X. Lei, X. Qian, and G. Zhao, "Rating Prediction Based on Social Sentiment From Textual Reviews," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1910–1921, 2016.

[13] C. Lin, Y. He, R. Everson, and S. Ruger, “Weakly Supervised Joint Sentiment-Topic Detection from Text,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 1134–1145, 2012.

[14] Y. Jo and A.-H. Oh, “Aspect and sentiment unification model for online review analysis,” in *Proceedings of the fourth ACM international conference on Web search and data mining - WSDM ’11*. ACM Press, 2011, pp. 815–824.

[15] H. Peng, Y. Ma, Y. Li, and E. Cambria, “Learning multi-grained aspect target sequence for Chinese sentiment analysis,” *Knowledge-Based Systems*, vol. 148, pp. 167–176, 2018.

[16] Y. Ma, H. Peng, and E. Cambria, “Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[17] S. Kim, J. Zhang, Z. Chen, A. Oh, and S. Liu, “A hierarchical aspect-sentiment model for online reviews,” in *Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013*, 2013, pp. 526–533.

[18] J. Serrano-Guerrero, F. Chiclana, J. A. Olivas, F. P. Romero, and E. Homapour, “A T1OWA fuzzy linguistic aggregation methodology for searching feature-based opinions,” *Knowledge-Based Systems*, vol. 189, p. 105131, 2019.

[19] M. Tsytsarau and T. Palpanas, “Survey on mining subjective data on the web,” *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 478–514, 2011.

[20] R. R. Yager, “On ordered weighted averaging aggregation operators in multicriteria decision making,” *IEEE transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[21] L. A. Zadeh, “A computational approach to fuzzy quantifiers in natural languages,” *Computers and Mathematics with Applications*, vol. 9, no. 1, pp. 149 – 18, 1983.

[22] R. R. Yager, “Quantifier guided aggregation using OWA operators,” *International Journal of Intelligent Systems*, vol. 11, no. 1, pp. 49–73, 1996.

[23] H. Wang, Y. Lu, and C. Zhai, “Latent aspect rating analysis on review text data,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’10*. ACM Press, 2010, pp. 783–792.

[24] V. Torra, “The weighted OWA operator,” *International Journal of Intelligent Systems*, vol. 12, no. 2, pp. 153–166, 1997.

[25] J. Merigo and A. M. Gil-Lafuente, “The induced generalized OWA operator,” *Information Sciences*, vol. 179, no. 6, pp. 729–741, 2009.

[26] R. R. Yager, “Including importances in OWA aggregations using fuzzy systems modeling,” *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 2, pp. 286 – 294, 1998.