A Joint Model of Feature Mining and Sentiment Analysis for Product Review Rating

Jorge Carrillo de Albornoz, Laura Plaza,
Pablo Gervás, and Alberto Díaz

Universidad Complutense de Madrid,
Departamento de Ingeniería del Software e Inteligencia Artificial,
Madrid, Spain
{jcalbornoz,lplazam,albertodiaz}@fdi.ucm.es,
pgervas@sip.ucm.es

Abstract. The information in customer reviews is of great interest to both companies and consumers. This information is usually presented as non-structured free-text so that automatically extracting and rating user opinions about a product is a challenging task. Moreover, this opinion highly depends on the product features on which the user judgments and impressions are expressed. Following this idea, our goal is to predict the overall rating of a product review based on the user opinion about the different product features that are evaluated in the review. To this end, the system first identifies the features that are relevant to consumers when evaluating a certain type of product, as well as the relative importance or *salience* of such features. The system then extracts from the review the user opinions about the different product features and quantifies such opinions. The salience of the different product features and the values that quantify the user opinions about them are used to construct a *Vector of Feature Intensities* which represents the review and will be the input to a machine learning model that classifies the review into different rating categories. Our method is evaluated over 1000 hotel reviews from *booking.com*. The results compare favorably with those achieved by other systems addressing similar evaluations.

Keywords: automatic product rating, feature mining, polarity detection, sentiment analysis.

1 Introduction and Background

During the last decade, product review forums have become commonplace, and an increasing number of websites provide platforms for customers to publicize their personal evaluations and opinions of products and services. The information in product reviews is of great interest to both companies and consumers. Companies and organizations spend a huge amount of money to find customers' opinions and sentiments, since this information is useful to exploit their marketing-mix in order to affect consumer satisfaction. Individuals are interested in others' opinions when purchasing a product or hiring a service. In fact, according to a

P. Clough et al. (Eds.): ECIR 2011, LNCS 6611, pp. 55–66, 2011.

survey of ComScore[1], online costumer-generated reviews have significant impact on purchase decision, so that consumers are willing to pay at least 20% more for services receiving an *Excellent*, or 5-star, rating than for the same service receiving a *Good*, or 4-star, rating.

This situation has raised many NLP challenges, commonly referred to as *Sentiment Analysis*, such as *subjectivity detection*, *polarity recognition* and *rating inference*. Subjectivity detection aims to discover subjective or neutral terms, phrases or sentences, and it is frequently used as a previous step in polarity and rating classification [1,2,3]. Polarity recognition attempts to classify texts into positive or negative [4,5,6]. The rating inference task goes a step further and tries to identify different degrees of positivity and negativity, e.g. *strongly-negative*, *weakly-negative*, *fair*, *weakly-positive* and *strongly-positive* [6,7,8,9].

Focusing on product review classification, various approaches have been proposed during the last decade. Most of them only consider the polarity of the opinions (i.e. negative *vs.* positive) and rely on machine learning (ML) techniques trained over vectors of linguistic feature frequencies. Pang et al. [4], for instance, present a comparison between three ML algorithms trained on the frequencies of positive and negative terms, and conclude that unigram-based SVM classifiers can be efficiently used in polarity classification of movie reviews. Martineau and Finin [10] use a similar approach on the same corpus where the words are scored using a Delta TF-IDF function before classifying the reviews into positive and negative. A more ambitious task is proposed by Brooke [7], whose goal is to classify reviews of different types of products into three and five rating classes, respectively, using a set of linguistic features including intensification, negation, modality and discourse structure. However, none of these approaches take into account other factors that affect the polarity of an opinion, and especially the strength of this polarity, such as the aspects or features on which the reviewer opinions are expressed and the relations between them. We hypothesize that humans have a conceptual model of what is relevant regarding a certain product or service that clearly influences the polarity and strength of their opinions. For instance, when evaluating a hotel, reviewers seem to be mainly concerned about its location, cleanliness, staff, etc; whereas other aspects, such as nearby shops and restaurants or the bed size, are less important. Therefore, we argue that, to successfully understand the user opinion about a product, it is necessary to combine feature mining and sentiment analysis strategies.

This assertion is not novel, others have noticed it [11,12,13]. Carenini and colleagues [11] present a system for summarizing evaluative arguments which relies on the detection of features of the entity that is evaluated. They use the association rule mining approach presented in [14] to obtain a first list of features. Since the number of features can be unmanageable (around 100-200 features per product), they use an *ad hoc* set of *User-Defined Features (UDF)* to reduce this list. Tivov and McDonald [12] propose a statistical model which is able to discover topics or rating aspects and to extract textual evidence from reviews supporting each of these ratings. They evaluate their approach on a corpus of

[1] ComScore, `http://www.comscore.com/`. Last visited on 15 October 2010.

hotel reviews from *TripAdvisor.com.* This approach has two main limitations: first, it needs a pre-defined set of aspects for the purpose of extraction, which also have to be accompanied by a user rating (e.g. *Food: 2; Decor: 1; Service: 3; Value: 2*). This information is not usually available in most review collections, where users usually give a unique score that represents their overall rate for the product along with a free-text describing their opinions about one or more product aspects. Second, their system describes the product aspects using expressions such as "great reception" or "helpful staff" for the aspect *Service.* In our opinion, the words "great" and "helpful" in the previous expressions should not be considered representative of the aspect *Service* of a hotel, but may affect other aspects (e.g. "great room" or "helpful shuttle service"). Kim and Hovy [13] present a system that automatically extracts the pros and cons from online reviews by finding the holder and the topic of the opinion. However, they do not quantify the strength of these pros and cons, nor they predict the overall rating of the reviews.

In this paper, we focus on measuring the polarity and strength of opinions, especially in those expressed in product reviews. We propose a model that leverages the user opinion on the different product features to predict the rating of a review. The model works in 4 phases. First, it identifies the features that are important to consumers when evaluating a certain type of product. Second, it locates in the review the sentences where the user opinions on the different product features are stated. Third, it computes the polarity and strength of the opinion expressed in each sentence. Finally, it computes a single score for each feature, based on the polarity of the sentences associated to it, and constructs a *Vector of Feature Intensities* which represents the review and will be the input to a machine learning algorithm that predicts a rating for the review.

Our approach improves previous work in three main points. First, it does not make use of any previous knowledge about the product features that are relevant to the user, but discovers them automatically from a set of reviews using an unsupervised model. This allows the system to be directly portable to new types of products and services. Second, the set of discovered features is small and meaningful enough for the user, but each feature is defined by a number of concepts able to accurately describe it, independently of the vocabulary used. Third, the system estimates the weight of each product feature in the overall user opinion to predict a more precise rating.

2 Data Collection: The *HotelReview* Corpus

We collected 25 reviews per hotel for 60 different hotels (1500 reviews) from *booking.com*[2]. Each review contains the following information:

- The city where the hotel is located, the reviewer nationality, the date when the review was written and the type of reviewer from a set of 7 categories, such as *solo traveler*, *young couple* and *group.*

[2] http://www.booking.com/

- A score in 0-10 describing the overall opinion of the reviewer. This score is not given by the reviewer, but automatically calculated by *booking.com* from the rates assigned by the reviewer to 5 aspects: *Hotel staff*, *Services/facilities*, *Cleanliness of your room*, *Comfort*, *Value for money* and *Location*. Unfortunately, these disaggregated scores are not available in the reviews.
- A brief free-text describing, separately, what the reviewer liked and disliked during the stay in the hotel.

We have observed that the overall score assigned to a review frequently bears no relation at all with the text describing the user opinion about the hotel, so that two reviews with nearly the same score may reflect very different opinions. For instance, the two following reviews are assigned the score '6.5', but the second is clearly more negative than the first:

- *Good location. Nice roof restaurant - (I have stayed in the baglioni more than 5 times before). Maybe reshaping/redecorating the lobby.*
- *Noisy due to road traffic. The room was extremely small. Parking awkward. Shower screen was broken and there was no bulb in the bedside light.*

To overcome this drawback, we asked two annotators to assign a first category within the set *[Excellent, Good, Fair, Poor, Very poor]* and a second category within the set *[Good, Fair, Poor]* to each review based on the text describing it. To solve inter-judge disagreement, all the reviews with conflicting judgments were removed. Finally, we randomly selected 1000 reviews[3]. The final distribution of the reviews is 200 for each class in the 5-classes categorization and 349, 292 and 359, respectively, in the 3-classes categorization. An example of hotel review is shown in Table 1.

Table 1. An example of hotel review from the *HotelReview* corpus

<**HotelReview** ***idDoc=***"D_8" ***hotelID=***"H_2" ***hotelLocation=***"Paris" ***reviewerCategory=***"Young couple" ***reviewerNationality=***"Belgium" ***date=***"February 10, 2010" ***score=***"9.3" ***5_classes_intensity=***"Good"> ***3_classes_intensity=***"Good">

<**PositiveOpinion**> I liked the location, breakfast was nice as well as Tea Time Buffet, that was really nice. Parking on weekends is free (on the street, and it's safe). We got to the room and it was very smelly (cigarettes) so we asked and changed and got a nice room. I'd recommend this hotel definitely. But hey... it's not a 4 star hotel...</**PositiveOpinion**>

<**NegativeOpinion**>Staff is nice except for one receptionist (a man) at night, he was not helpful at all, I asked him for directions and he said there's nothing I can do if you don't know Paris. Anyways, everybody else was nice.</**NegativeOpinion**>
</**HotelReview**>

3 Automatic Product Review Rating

In this section we present a novel approach to product review rating. The method is based on identifying the features of concern to the consumers of a product,

[3] This collection is available for research purposes.
`http://nil.fdi.ucm.es/index.php?q=node/456`

extracting the product features that have been commented on by the reviewer, and weighting the comments on each product feature to estimate the overall sentiment of the reviewer about the product.

3.1 Step I: Detecting Salient Product Features

The aim of this step is to identify the features that are relevant to consumers when evaluating products of a certain type, as well as the relative importance or *salience* of such features. To this end, we adapt the summarization method presented in [15], which we explain here for completeness.

Given a set of reviews for a number of products of the same type, we first apply a shallow pre-processing over the text describing the users' opinions, including POS tagging and removing stopwords and high frequency terms. We next translate the text into WordNet concepts using the *lesk* algorithm [16] to disambiguate the meaning of each term according to its context. After that, the WordNet concepts for nouns are extended with their hypernyms, building a graph where the vertices represent distinct concepts in the text and the edges represent *is-a* relations between them. Our experimental results have shown that the use of verbs in this graph includes very general information that negatively affects the rating prediction step. Regarding adjectives and adverbs, we do not consider words from these gramatical categories to represent the product features, but to express the user opinions about them.

We next expand the graph with a semantic similarity relation, so that a new edge is added that links every pair of leaf vertices whose similarity exceeds a certain threshold. To calculate this similarity, different measures have been tested. Finally, each edge is assigned a weight in [0,1]. This weight is calculated as the ratio between the relative positions in their hierarchies of the concepts linked by the edge.

The vertices are next ranked according to their salience or prestige. The *salience* of a vertex, v_i, is calculated as the sum of the weight of the edges connected to it multiplied by the frequency of the concept represented by v_i in the set of reviews. The top n vertices are grouped into *Hub Vertex Sets (HVS)* [17], which represent sets of concepts strongly related in meaning. A degree-based clustering method is then run over the graph to obtain a non-predefined number of clusters, where the concepts belonging to the HVS represent the centroids. The working hypothesis is that each of these clusters represents a different product feature. Figure 1a shows the highest salience concepts or centroid for each of the 18 feature clusters generated from a set of 1500 hotel reviews from *booking.com* using the Jiang and Conrath [18] similarity measure and a 0.25 similarity threshold to build the graph. In Figure 1b, all the concepts belonging to the feature cluster **"room"** are shown.

3.2 Step II: Extracting the User Opinion on Each Product Feature

Once the system knows the product features or aspects that are of concern to consumers, the next step is to extract from the review the opinions expressed on

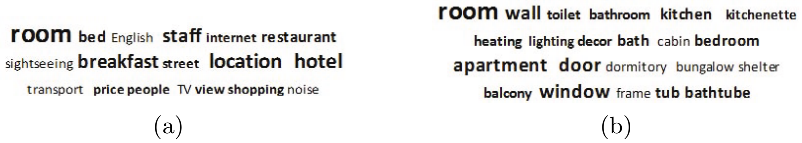
room bed English staff internet restaurant
sightseeing breakfast street location hotel
transport price people TV view shopping noise
(a)
room wall toilet bathroom kitchen kitchenette
heating lighting decor bath cabin bedroom
apartment door dormitory bungalow shelter
balcony window frame tub bathtube
(b)

Fig. 1. (a) Highest salience concept for each product feature. (b) Concepts belonging to feature **room**. A bigger letter size indicates a higher salience.

such features. Thus, we need to locate in the review all textual mentions related to each product feature. To do this, we map the reviews to WordNet concepts in the same way than in the previous step, and we associate the sentences in the review to the product features they refer to using three heuristics:

- ***Most Common Feature (MCF):*** The sentence is associated to the feature with which it has more WordNet concepts in common.
- ***All Common Features (ACF):*** Since a sentence may contain information related to different features, we associate the sentence to every feature with some concept in common.
- ***Most Salient Feature (MSF):*** For each feature and sentence, we compute a score by adding the salience of the concepts in the sentence that are also found in the feature cluster. Then, the sentence is associated to the highest score feature.

It must be noted that a sentence may consist only of concepts not included in any feature cluster, so that it cannot be associated to any of them. To avoid losing the information in these sentences we create a further cluster (*other features*), and associate these sentences to it[4].

3.3 Step III: Quantifying the User Opinions

We next aim to quantify the opinion expressed by the reviewer on the different product features. To this end, we predict the polarity of the sentences associated to each feature. Since it is unlikely that a user will annotate every sentence in a review as being positive or negative, we use the polarity recognition system presented in Carrillo de Albornoz et al. [9]. The main idea of this method is to extract the WordNet concepts in a sentence that entail an emotional meaning, assign them an emotion within a set of categories from an affective lexicon, and use this information as the input of a logistic regression model to predict the polarity of the sentence and the probability of this polarity. The main points of this approach, as pointed by the authors, are: (1) the use of WordNet and a word sense disambiguation algorithm, which allows the system to work with concepts rather than terms, (2) the use of emotional categories instead of terms as classification attributes, and (3) the use of negations and quantifiers to invert, increase or dismiss the intensity of these emotions. This system has been shown to outperform previous systems which aim to solve the same task.

[4] We tried ignoring these sentences and found it to be less effective.

For instance, when the system is run over the sentence "*In the room, there were no problems with the heating system, so even if outside it was quite freezing, in the hotel was warm enough*", the sentence is classified as positive with a probability of *0.963*.

3.4 Step IV: Predicting the Rating of a Review

Once all the relevant product features are extracted, and the user opinions on each feature are quantified, the system should aggregate this information to provide an average rating for the review (e.g. *Good*, *Fair* and *Poor*). We translate the product review into a *Vector of Feature Intensities (VFI).* A VFI is a vector of *N+1* values, each one representing a different product feature and the *other features.* We experiment with two strategies for assigning values to the VFI positions:

- ***Binary Polarity (BP):*** For each sentence in the review, the position of the feature to which the sentence has been assigned is increased or decreased by 1, depending on whether the sentence was predicted as positive or negative.
- ***Probability of Polarity (PP):*** Similar to the previous one, but the feature position is increased or decreased by the probability of the polarity assigned to the sentence by the polarity classifier.

For instance, the review shown in Table 1 will produce the following *VFI* when the *MSF* heuristic and the *BP* strategy are used, and the set of features in Figure 1a is considered: _[**-1.0**, 0.0, 0.0, **0.0**, 0.0, 0.0, 0.0, 0.0, **-1.0**, 0.0, 0.0, **1.0**, 0.0, **1.0**, 0.0, 0.0, 0.0, 0.0, **1.0**]._ In this review, the sentence *"I liked the location..."* is labeled by the polarity classifier as positive, and assigned to the feature represented by the concept **location** (12th position in the *VFI*). Even if this sentence contains information related to concepts from other features (e.g. breakfast and buffet, from feature **breakfast**), location presents a higher salience and so the sentence is assigned to it, adding *1.0* to its vector position. Similarly, the sentence *"Parking on weekends..."* is labeled as positive and assigned to the **street** feature (14th position in the *VFI*). The sentence *"We got to the room..."* is labeled as negative and assigned to the feature **room** (1st position in the *VFI*), decreasing this position in *1.0.* In turn, the sentences *"I'd recommend this hotel..."* and ‘*‘But hey... it's not...”* are both assigned to the feature **hotel** (4th position in the *VFI*) but, since the first one is classified as positive and the last one is classified as negative, their scores are neutralized. Finally, the sentence *"Staff is nice except for..."* is assigned to the **staff** feature (9th position in the *VFI*) with a negative intensity, and the sentence *"Anyways, everybody else was nice"* is assigned to the **other features** (the last position in the *VFI*) with a positive intensity.

The VFI is finally used as the input to a machine learning algorithm that classifies the review into different rating categories.

4 Evaluation

4.1 Evaluation Setup

We use the *HotelReview* collection described in Section 2 for evaluating the method. This collection contains 1000 reviews manually labeled within two different sets of categories: *[Good, Fair, Poor]* and *[Excellent, Good, Fair, Poor, Very Poor]*. To determine the best ML algorithm for review rating, several Weka classifiers were trained and evaluated using 10-fold cross validation. We only show the results of the three best performance classifiers: a logistic regression model (*Logistic*), a support vector machine (*LibSVM*) and a functional tree (*FT*). Furthermore, since the sentences in the reviews in the *HotelReview* corpus are labeled with either a positive or negative polarity, depending on whether they appear in the <*PositiveOpinion*> or <*NegativeOpinion*> section of the review, we use these labeled sentences to train the polarity classifier described in Section 3.3, again using 10-fold cross validation.

We test the system using different hotel feature sets. As explained in Section 3.1, these sets depend on three parameters: the set of reviews, the similarity measure and the similarity threshold used to build the graph. We have experimented using 50, 1000 and 1500 reviews randomly selected from *booking.com*; three different similarity measures (Lesk [16], Jiang & Conrath [18], and Lin [19]); and various similarity thresholds from 0.1 to 0.5. We have found that the best results are achieved when the Jiang & Conrath similarity is used and the similarity threshold is set to 0.25. Thus, we only report here the results for the three feature sets obtained using this similarity and threshold: ***Feature set 1***, which is built from 50 reviews and consists of 24 features and 114 concepts; ***Feature set 2***, which is built from 1000 reviews and consists of 18 features and 330 concepts; and ***Feature set 3***, which is built from 1500 reviews and consists of 18 features and 353 concepts.

We also compare our system with two state-of-art approaches. The first one is the SVM over bags of unigrams approach presented in Pang et al. [4]. Let $u_1, ..., u_m$ be a set of m unigrams that can appear in a review. Let $f_i(r)$ be the number of times u_i occurs in the review r. Then, r is represented by the review vector $R = f_1(r), ..., f_m(r)$. The set of m unigrams is limited to unigrams appearing at least 4 times in the 1000-review corpus. In spite of its simplicity, this approach turned out to be more effective than more complex approaches using features such as bigrams, part-of-speech tagging and word positions, and might actually be considered somewhat difficult to beat. The second approach is just the Carrillo de Albornoz et al. [9] algorithm for sentence polarity prediction presented in Section 3.3. In order to work with reviews rather than sentences, the whole text in a review is considered as a unique sentence.

4.2 Results

We first examine the effect of the product feature set on the review classification. Table 2 shows the accuracy for three Weka classifiers in a 3-classes task (i.e. *Good*, *Fair* and *Poor*), using the three feature sets presented in the previous section. For these experiments, we use the *Binary Polarity* strategy for assigning values to the VFI attributes, as explained in Section 3.4.

Table 2. Average accuracies for different classifiers, using different feature sets and sentence-to-feature assignment strategies

Method	Feature Set 1			Feature Set 2			Feature Set 3		
	MCF	*ACF*	*MSF*	*MCF*	*ACF*	*MSF*	*MCF*	*ACF*	*MSF*
Logistic	69.8	67.7	69.8	70.4	67.4	***70.8***	69.1	67.4	70
LibSVM	69	67.1	69.2	69	67.8	***69.2***	68.8	67.7	69
FT	66.8	64.2	66.8	66.3	65.2	***68.6***	68.4	65.8	68.4

As shown in Table 2, the *Feature set 2* reports the best results for all classifiers. However, the accuracy differs little across different feature sets, which seems to indicate that increasing the number of reviews used for extracting the features does not necessarily improve the accuracy of rating prediction. As a result of this finding, we use the *Feature set 2* in the remaining experiments.

We also aim to determine which of the three heuristics for sentence-to-feature assignment produces the best outcome (see Section 3.2). As it may be seen from Table 2, the *MSF* heuristic reports the best results for most algorithms and feature sets, but the *MCF* also reports very close accuracies. In contrast, the *ACF* heuristic produces significantly worse results. Although intuitively any information about a product feature in a sentence should be taken into account by the classifier, these results seem to indicate that only the main feature in each sentence provides useful information for the rating prediction task. On the other hand, we have observed that the heuristics *MCF* and *MSF* produce very similar sentence-to-feature mappings, which leads to similar classification results.

We next examine the use of the *Probability of Polarity* strategy for assigning values to the *VFI* (see Section 3.4). As shown in Table 3, the use of this strategy improves the average accuracy for all ML algorithms. The best performance (71.7%) is achieved by *Logistic*, increasing its accuracy by as much as 0.9% beyond that of the *Binary Polarity* strategy (Table 2). We presume that this probability of polarity somehow captures the degree of negativity/positivity of a sentence, which results in useful information for the classifier, since it is clearly not the same to say *“The bedcover was a bit dirty”* than *“The bedcover was terribly dirty”*. Finally, it may be also observed that the results produced by the system in all ML techniques significantly outperform those of the *Pang et al. [4]* and *Carrillo de Albornoz et al. [9]* approaches.

We finally tested our approach in a 5-classes prediction task (e.g. *Excellent*, *Good*, *Fair*, *Poor* and *Very Poor*). The results (Table 4) considerably decrease with respect to the 3-classes task, but are still significantly better than those achieved by the *Pang et al.* [4] and *Carrillo de Albornoz et al.* [9] approaches. This result was expected, since it is a more difficult task. However, to find out further reasons for this decrease, we examined the confusion matrix and discovered that most classification errors come from *Good* and *Poor* instances which are classified

Table 3. Average results for different classifiers in the 3-classes prediction task

Method	Acc.	Good		Fair		Poor	
		Pr.	Re.	Pr.	Re.	Pr.	Re.
Logistic	**71.7**	77.3	82.5	58.6	46.6	74	81.7
LibSVM	69.4	73.6	83	57.2	38	71.1	81.7
FT	66.9	73	76.9	50.8	43.2	71.4	76.5
Carrillo de Albornoz et al. [9]	66.7	71.7	82.7	48.7	32.9	70.4	78.5
Pang et al. [4]	54.2	63.6	61.8	38.7	39.7	58.1	58.5

Table 4. Average results for different classifiers in the 5-classes prediction task

Method	Acc.	Excellent		Good		Fair		Poor		Very Poor	
		Pr.	Re.	Pr.	Re.	Pr.	Re.	Pr.	Re.	Pr.	Re.
Logistic	**46.9**	52.6	65	39.9	30.5	38.3	38.5	41.7	40	58.5	60.5
LibSVM	45.3	52.3	68	33.1	20.5	37.4	36.5	37.3	40.5	59.8	61
FT	43.7	49	59.5	27.6	18.5	37.6	37	39.7	35.5	55.1	68
Carrillo de Albornoz et al. [9]	43.2	51.4	81.5	38.7	23	40.9	13.5	31.6	55.5	57.8	42.5
Pang et al. [4]	33.5	59.7	46	26.3	39	29.9	26	26.5	28	34.8	28.5

as *Excellent* and *Very Poor* respectively. We also checked that most problematic instances correspond to borderline cases where the final classification involves some degree of subjectivity from human taggers. For instance, the two following reviews are classified in the corpus as *Very Poor* and *Poor* respectively, but have been considered by different judges to entail a similar degree of negativity:

- *Only the location was a positive aspect. The room was very small, in the basement of the hotel and we could hear people walking around all night long. The shower and the toilet closets were really uncomfortable and small as well.*
- *Very near transport to London central. On arrival the apartment was very smelly of old spicy food and standard of cleanliness was poor. Entrance to apartment (hallway) needs a paint job.*

5 Discussion and Conclusions

Our experimental results show that our feature-driven approach to product review rating performs significantly better than previous approaches, which confirms our intuition that the different product features have different impact on the user opinion about a product. Our approach succeeds in identifying salient features which can be easily obtained from a relatively small set of product reviews and are quite independent of the reviews used to extract them. We speculate that this indicates that users are concerned about a relatively small set of product features which are also quite consistent among users. Also, the identification of salient features is carried out without previous knowledge, so the system may be easily ported to other domains. We have also observed that the differences between the various Weka classifiers are not marked, which suggests that the proposed data representation properly captures the salient product features and the user opinions about them.

On the other hand, since we use the polarity classifier presented in Carrillo de Albornoz et al. [9] to quantify the opinion of the users about each feature, the error of this classifier increases the error of our system. To estimate the effect of error propagation, we repeat the experiments reported in Table 3, but using the sentence polarity categorization as given in the review (i.e. all sentences within the tags <*PositiveOpinion*> are considered positive and all sentences within the tags <*NegativeOpinion*> are considered negative). As a result, we improve accuracy to 84% for *Logistic*, 83% for *LibSVM* and 81.9% for *FT*. This improvement seems to indicate that a good amount of sentences are incorrectly classified. We believe the error in this classifier mainly comes from: (1) mislabeled instances in the training set (e.g. the sentence *"Anyway, everybody else was nice"* in the review shown in Table 1 is incorrectly placed in the <*NegativeOpinion*> section), (2) very frequent spelling errors in the reviews and (3) the presence of *neutral* sentences that do not express any opinion but are necessarily classified as positives or negatives. We also plan to adapt the polarity classifier to take into account the features when evaluating the polarity in order to estimate how this could affect the rating classifier.

However, most classification errors come from reviews in which it is not possible to assign any sentence to any feature and, as a result, all sentences are assigned to the *other features*. This situation occurs when the review only deals with secondary features, but especially when the method is not able to determine the focus of the sentences, so that it cannot decide which feature a sentence is talking about. For example, the review *"Dirty. Stinky. Unfriendly. Noisy."* presents 4 sentences that are assigned to the *other features*, but should be assigned to different features if the method could identify the objects/features to which the four adjectives refer to. The same occurs in the sentence *"Anyway, everybody else was nice"* from the review in Table 1, where the system does not know that 'everybody' refers to the hotel staff. This problem has been previously noted by Pang et al. [4] and Turney [5] and is regarded as a kind of co-reference problem. We will address this issue in the near future.

To end with, in the long term future we plan to apply the method to deal with documents in Spanish language, using the Spanish version of WordNet available in EuroWordNet.

Acknowledgments

This research is funded by the Spanish Government through projects TIN2009-14659-C03-01 and TSI-020100-2009-252, and the FPU program; and the Comunidad Autónoma de Madrid and the ESF through the IV PRICIT program.

References

1. Wiebe, J.M., Bruce, R.F., O'Hara, T.P.: Development and use of a gold-standard data set for subjectivity classification. In: Proc. of ACL, pp. 246–253 (1999)
2. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proc. of ACL, pp. 271–278 (2004)
3. Kim, S.M., Hovy, E.: Determining the sentiment of opinions. In: Proc. of COLING, pp. 1367–1373 (2004)
4. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using Machine Learning techniques. In: Proc. of CoRR (2002)
5. Turney, P.D.: Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proc. of ACL, pp. 417–424 (2002)
6. Esuli, A., Sebastiani, F.: Determining term subjectivity and term orientation for opinion mining. In: Proc. of EACL, pp. 193–200 (2006)
7. Brooke, J.: A semantic approach to automated text sentiment analysis. PhD thesis, Simon Fraser University (2009)
8. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity: an exploration of features for phrase-level sentiment analysis. Computational Linguistics 35(3) (2009)
9. Carrillo de Albornoz, J., Plaza, L., Gervás, P.: A hybrid approach to emotional sentence polarity and intensity classification. In: Proc. of 14th CoNLL, pp. 153–161 (2010)
10. Martineau, J., Finin, T.: Delta TF-IDF: An improved feature space for sentiment analysis. In: Proc. of 3rd AAAI Conference on Weblogs and Social Media (2009)
11. Carenini, G., Ng, R.T., Pauls, A.: Multi-document summarization of evaluative text. In: Proc. of EACL (2006)
12. Titov, I., McDonald, R.: A joint model of text and aspect ratings for sentiment summarization. In: Proc. of ACL/HLT, pp. 308–316 (2008)
13. Kim, S.M., Hovy, E.: Automatic identification of pro and con reasons in online reviews. In: Proc. of COLING/ACL, pp. 483–490 (2006)
14. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: Proc. of AAAI (2004)
15. Plaza, L., Díaz, A., Gervás, P.: Automatic summarization of news using wordnet concept graphs. IADIS International Journal on Computer Science and Information System V, 45–57 (2010)
16. Lesk, M.E.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In: Proc. of SIGDOC (1986)
17. Yoo, I., Hu, X., Song, I.Y.: A coherent graph-based semantic clustering and summarization approach for biomedical literature and a new summarization evaluation method. BMC Bioinformatics 8(9) (2007)
18. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proc. of International Conference Research on Computational Linguistics (1997)
19. Lin, D.: An information-theoretic definition of similarity. In: Proc. of ICML, pp. 296–304 (1998)