Multi-Facet Rating of Product Reviews

Stefano Baccianella, Andrea Esuli and Fabrizio Sebastiani

Istituto di Scienza e Tecnologia dell'Informazione
Consiglio Nazionale delle Ricerche
Via Giuseppe Moruzzi 1 – 56124 Pisa, Italy
{*firstname.lastname*}@isti.cnr.it

Abstract. Online product reviews are becoming increasingly available, and are being used more and more frequently by consumers in order to choose among competing products. Tools that rank competing products in terms of the satisfaction of consumers that have purchased the product before, are thus also becoming popular. We tackle the problem of rating (i.e., attributing a numerical score of satisfaction to) consumer reviews based on their textual content. We here focus on *multi-facet* review rating, i.e., on the case in which the review of a product (e.g., a hotel) must be rated several times, according to several aspects of the product (for a hotel: cleanliness, centrality of location, etc.). We explore several aspects of the problem, with special emphasis on how to generate vectorial representations of the text by means of POS tagging, sentiment analysis, and feature selection for ordinal regression learning. We present the results of experiments conducted on a dataset of more than 15,000 reviews that we have crawled from a popular hotel review site.

1 Introduction

Online product reviews are becoming increasingly available across a variety of Web sites, and are being used more and more frequently by consumers in order to make purchase decisions from among competing products[1]. For example, according to a study [1] performed on TripAdvisor[2], one of the most popular online review sites for tourism-related activities, among the users that use the TripAdvisor online booking system 97.7% are influenced by other travelers' reviews, and among them 77.9% use the reviews as a help to choose the best place to stay.

Software tools that organize product reviews and make them easily accessible to prospective customers are thus going to be more and more popular. Among the issues that the designers of these tools need to address are (a) content aggregation, such as in pulling together reviews from sources as disparate as newsgroups, blogs, and community Web sites; (b) content validation, as in filtering out fake reviews authored by people with vested interests [2]; and (c) content organization, as in automatically ranking competing products in terms of the satisfaction of consumers that have purchased the product before.

[1] http://dataforbreakfast.com/?p=115
[2] http://www.tripadvisor.com/

We address a problem related to issue (c), namely, rating (i.e., attributing a numerical score of satisfaction to) consumer reviews based on their textual content. This problem arises from the fact that, while some online product reviews consist of a textual evaluation of the product *and* a score expressed on some ordered scale of values, many other reviews contain a textual part only. These latter reviews are difficult for an automated system to manage, especially when a qualitative comparison among them is needed in order to determine whether product x is better than product y, or to identify the best product in the lot. Tools capable of interpreting a text-only product review and scoring it according to how positive the review is, are thus of the utmost importance.

In particular, our work addresses the problem of rating a review when the value to be attached to it must range on an *ordinal* (i.e., discrete) scale. This scale may be in the form either of an ordered set of *numerical* values (e.g., one to five "stars"), or of an ordered set of *non-numerical* labels (such as e.g., Poor, Good, Very good, Excellent); the only difference between these two cases is that, while in the former case the distances between consecutive scores are known, this is not true in the latter case. We also focus on *multi-facet* rating of product reviews, i.e., on the case in which the review of a product (e.g., a hotel) must be rated several times, according to several orthogonal aspects of the product (for a hotel: cleanliness, centrality of location, etc.).

The system we have realized could work as a building block for other larger systems that implement more complex functionality. For instance, a Web site containing product reviews whose users only seldom rate their own reviews could use this system to learn from the rated reviews to rate the others; yet another Web site containing only unrated product reviews could learn, from the rated reviews of some other site which contains rated reviews, to rate its own reviews.

This work mostly focuses, rather than on the learning device used for generating a review rater, on the generation of the vectorial representations of the reviews that must be given as input to the learning device. These representations cannot simply consist of the usual bag-of-words representations used in classifying texts *by topic*, since classifying texts *by opinion* (which is the key contents of reviews) requires much subtler means [3]. Two expressions such as "A great hotel in a horrible town!" and "A horrible hotel in a great town!" would receive identical bag-of-words representations, while expressing opposite opinions on the hotel. We have focused on three aspects of the generation of meaningful representations of product reviews: (a) the extraction of complex features based on patterns of parts of speech; (b) making the extracted features more robust through the use of a lexicon of opinion-laden words; and (c) the selection of discriminating features through techniques explicitly devised for ordinal regression (an issue which had practically received no attention in the literature).

The rest of the paper is organized as follows. Section 2 describes the key part of our work, i.e., how we generate the vectorial representations of the reviews. Section 3 describes a hotel review dataset we have crawled from the Web and the results of the experiments we have run on it. Section 4 presents related work, while Section 5 concludes, discussing avenues for future research.

2 Generating vectorial representations of product reviews

In machine learning the problem of rating data items with values ranging on an ordinal scale is called *ordinal regression* (OR). OR consists in estimating a *target function* $\Phi : X \rightarrow Y$ which maps each object $x_j \in X$ into exactly one of an ordered sequence $Y = \langle y_1 \prec \ldots \prec y_n \rangle$ of *labels* (aka "scores", or "ranks"), by means of a function $\hat{\Phi}$ called the *classifier*[3]. This problem lies in-between *single-label classification*, in which Y is instead an unordered set, and *metric regression*, in which Y is instead a continuous, totally ordered set (typically: the set $\mathbb{R}$ of the reals). Throughout this work, as a learning device for ordinal regression we use *ϵ-support vector regression* (ϵ-SVR) [4], as implemented in the freely available LibSvm library [5]; we have set all its parameters at their default values.

As all supervised learning devices, ϵ-SVR requires all training and test examples to be represented as feature vectors. As a baseline representation we use bag-of words with cosine-normalized $tfidf$ weighting. As mentioned in the introduction, this representation cannot account for the subtle ways in which opinions are represented. In the rest of this section we will thus discuss our efforts at devising better representations for the purpose of product review rating.

2.1 Pattern extraction

Our first move away from the simplistic bag-of-words representation has consisted in spotting units of text larger than words that have the potential to be useful additional features. For instance, for distinguishing "A great hotel in a horrible town!" from "A horrible hotel in a great town!", it may be useful to use "great hotel" and "horrible hotel" as features in their own right. While most previous works on identifying indexing units larger than words have used frequency considerations alone (see e.g., [6]), we have chosen to bring to bear syntax; for instance, both "great hotel" and "horrible hotel" follow the part-of-speech (POS) pattern "JJ NN", where "JJ" stands for "adjective" and "NN" for "noun". We have thus defined three POS patterns (which we have creatively called A, B, C) which we deemed could identify meaningful larger-than-word units to be be used as features; see Figure 1 for a detailed grammar of these patterns. Note that we will use the expressions matching these patterns as features *additional* to the features extracted via bag-of-words; that is, if "horrible hotel" is extracted by one of the patterns we have defined, both "horrible", "hotel", and "horrible hotel" will be features of our vectorial representation.

Pattern A models (possibly complex) noun phrases, such as "nice room" or "very rude staff". Pattern B captures instead complex expressions that also contain a verb, such as "hotel was very nice" or "staff helped very much". Pattern C instead addresses expressions stating that a subject has or does not have some property, such as "has a nice restaurant" or "has a bar".

Different expressions we extract may state in different forms the same opinion about the same subject: for example, the type-B expression "the room was very

[3] Consistently with most mathematical literature we use the caret symbol (ˆ) to indicate estimation.

PATTERN	::= A | B | C	AP	Determiner/pronoun
A	::= [AT] ADJ NOUN	AT	Article
B	::= NOUN VERB ADJ	Be	Verb "to be"
C	::= Hv A	CC,CS	Conjunction
NOUN	::= [AT] [NN$] NN	Hv	Verb "to have"
ADJ	::= [CONG] ADV ADJ	JJ	Adjective
ADV	::= RB ADV | QL ADV | JJ | AP ADV | ϵ	NN,NN$	Noun and noun followed by Saxon genitive
CONG	::= CC | CS	QL	Qualifier
VERB	::= V | Be	RB	Adverb
		V	Verb (other than "be", "have", and "do")

Fig. 1. POS patterns used to extract larger-than-word units. The leftmost part of this table defines the three POS patterns, while the rightmost part lists the terminal symbols used in the leftmost part, as extracted by a standard POS tagger.

nice but small" and the type-A expression "very nice but small room" convey the same information, which is also the same information collectively conveyed by the two type-A expressions "very nice room" and "small room". We have thus defined two *canonical forms* in which the expressions matching our patterns are converted once extracted from text, with the double aim of (a) reducing the number of distinct but semantically equivalent features, and (b) increasing the statistical robustness of the remaining features by increasing their counts. The two canonical forms are "ADJ NN" (for A- and B-type expressions) and "HV ADJ NN" (for C-type expressions). The transformation of expressions into their corresponding canonical form is obtained by (i) removing articles ("the hotel was very nice and good located" $\mapsto$ "hotel was very nice and good located")[4]; (ii) splitting conjunctions, creating a pattern for every adjectival form ("hotel was very nice and good located" $\mapsto$ "hotel was very nice" + "hotel was good located"); (iii) removing auxiliary verbs ("hotel was very nice" $\mapsto$ "hotel very nice") (Applied only on Pattern B); (iv) putting adjectives in front of nouns ("hotel very nice" $\mapsto$ "very nice hotel").

POS tagging also provides information about the presence of negations. This allowed us to add an explicit negation in front of any expression for which the POS tagger detected the presence of a negation (e.g., "the staff was not nice" $\mapsto$ "not nice staff"), so as to avoid collapsing negated and non-negated statements of the same fact into the same feature.

Figure 2 shows a sample review from the training set of the corpus described in Section 3.1, with the expressions matching our POS patterns in boldface.

2.2 Pattern aggregation through sentiment analysis

In the expressions extracted so far, different opinion-bearing terms may be used to express sentiment of similar polarity (i.e., positive vs. negative) and strength. For example, both "horrible location" and "disgusting location" express a strongly negative feeling about the location of a hotel. We use a lexical resource of opinion-laden terms with the aim of mapping specific expressions conveying opinion (such as "disgusting location") into more "abstract" expressions

[4] Any syntactic mistake or clumsy English expression in the examples we use is genuine, i.e., it appears somewhere in our review dataset.

"**Great location**"! We loved the location of this hotel **the area was great** for **affordable restaurants**, bakeries, **small grocers** and near **several good restaurants**. Do not overlook the **lovely church** next door quite a treat! **The rooms were servicable** and some seemed to have been more recently refurbished. Just stay away from room 54 for the money it was a suite **the comfort was not worth** the price, **poor heater** and **horrible shower**, not a single shelf in the bathroom to hold a bar of soap. But 38 also a suite was much nicer. **The basic twin rooms were fine and small** as to be expected. I recommend this hotel overall but do not expect much help from the front desk as all but one of the staff bordered on surly. That was the most disappointing aspect of this **otherwise nice hotel**, **the breakfast was fine** and the breakfast **room was lovely**.

Fig. 2. An example hotel review from the dataset of Section 3.1. The expressions matching our POS patterns are shown in **boldface**.

Expression	Simple GI Expression	Enriched GI Expression
great location	[Positive] location	[Strong] [Positive] location
great hotel	[Positive] hotel	[Strong] [Positive] hotel
helpful staff	[Positive] staff	[Virtue] [Positive] staff
friendly staff	[Positive] staff	[Emot] [Virtue] [Positive] staff
good location	[Positive] location	[Virtue] [Positive] location
nice hotel	[Positive] hotel	[Virtue] [Positive] hotel
very helpful staff	[Positive] staff	very [Virtue] [Positive] staff
very friendly staff	[Positive] staff	very [Emot] [Virtue] [Positive] staff
excellent location	[Positive] location	[Virtue] [Positive] location
great place	[Positive] place	[Strong] [Positive] place

Fig. 3. The 10 most frequent expressions in the "Value" dataset (see Section 3.1), together with their corresponding simple and enriched GI expressions.

(such as "[Negative] location"). We then use these abstract expressions (here called *simple GI expressions*) as *additional* features for our vectorial representation (i.e., we retain as features both "horrible location", "disgusting location", and "[Negative] location"). The lexical resource we have chosen for our experiments is the [Positive]/[Negative] subset of the General Inquirer (GI) [7], a set of 1,915 (resp., 2,291) terms marked as having a positive (resp., negative) polarity. Examples of positive terms are "advantage", "fidelity" and "worthy", while examples of negative terms are "badly", "cancer", and "stagnant". In order to generate simple GI expressions, we match all the words in each of the extracted expressions against the GI lexicon[5] and, if the word is present, its [Positive] or [Negative] tag is used to generate a new expression in which the tag replaces the word (see Table 3 for examples).

In the GI, words are also marked according to an additional, finer-grained set of sentiment-related tags (see Figure 4); some of them denote the magnitude of the sentiment associated to the word, while others denote specific emotions and feelings evoked by the word. This allows us to cover the sentiment-carrying expressions that occur in our reviews in a finer-grained way. We thus generate a further type of expressions, which we call *enriched GI expressions*, by adding to all simple GI expressions the appropriate finer-grained sentiment-related tags. Table 3 reports the 10 most frequent expressions in the "Value" dataset (see Section 3.1) with the simple and enriched GI expressions that are generated from them. All enriched GI expressions are added to the feature set.

[5] For some words with multiple senses GI has more than one entry; we do not perform any word sense disambiguation, and thus simply choose the most frequent sense.

Tag	Description
[Strong]	words implying strength
[Power]	indicating a concern with power, control or authority
[Weak]	words implying weakness
[Submit]	connoting submission to authority or power, dependence on others, vulnerability to others, or withdrawal
[Pleasur]	words indicating the enjoyment of a feeling, including words indicating confidence, interest and commitment
[Pain]	words indicating suffering, lack of confidence, or commitment
[Feel]	words describing particular feelings, including gratitude, apathy, and optimism, not those of pain or pleasure
[Arousal]	words indicating excitation, aside from pleasures or pains, but including arousal of affiliation and hostility
[Emot]	words related to emotion that are used as a disambiguation category, but also available for general use
[Virtue]	words indicating an assessment of moral approval or good fortune, especially from the perspective of middle-class society
[Vice]	words indicating an assessment of moral disapproval or misfortune
[NegAff]	words of negative affect "denoting negative feelings and emotional rejection"
[PosAff]	words of positive affect "denoting positive feelings, acceptance, appreciation and emotional support"

Fig. 4. Fine-grained set of GI sentiment-related tags and their textual definitions.

2.3 Feature selection for ordinal regression

The final feature set thus consists of all words, all expressions (as from the patterns of Section 2.1), all simple GI expressions, and all enriched GI expressions. This means that the dimensionality of the resulting vector space may be very large. It seems thus necessary to add a feature selection phase, with the twofold aim of improving the efficiency of the learning phase and removing non-discriminating features. As in practically all text learning tasks we will follow a "filter" approach [8], according to which each candidate feature, irrespectively of its nature (word, pattern, etc.), is scored by a function that measures its discriminative power; only the t highest-scoring features will be retained.

There are many standard feature selection methods for text classification [9] and for metric regression [10]; on the other hand, research on feature selection for ordinal regression has been much scarcer, and to the best of our knowledge the only work which addresses this problem is [11]. . However, this method is not applicable in our context, since it amounts to classifying the training instances using the feature alone, evaluating the performance in terms of the chosen evaluation measure, and then taking the result as the importance score of the feature; since this amounts to learning a classifier for each feature, this method is applicable only when the original set of features is very small. In this work we propose and compare two pattern selection methods for ordinal regression that draw inspiration from work on feature selection for text classification.

Our first method, that we call *minimum variance* (MV), is based on measuring the variance of the distribution of a feature across the labels of our ordered scale, and retaining only the t features that have the smallest variance. For the purpose of computing variance, the labels are mapped to the first n natural numbers, and the value of a term occurrence is the natural number associated to the label of the document in which the term occurs. The intuitive justification of

MV is that a useful feature is one that is capable of discriminating a small portion of the ordered scale from the rest, and that features with a small variance are those which satisfy this property.

Our second method is inspired by [13], and is based on the observation that MV might well select many features that discriminate well *some* of the labels, while selecting few or no features that discriminate well the other labels. If, by absurd, all texts with label y were in German and all the other texts were in English, MV would likely pick mostly or only German words, since their variance is 0, with the consequence that an accurate model would likely be learned for y but not for the other labels. A solution to this problem is to perform feature scoring separately for each label (in a one-against-all fashion), and then to determine the final feature ranking with a round-robin algorithm in which the labels take turns in picking their most discriminating features. Our method (here called RRMV) thus consists in applying a RR policy to MV. In this specific version the features are "assigned" to the label closest to their average label value; then n distinct feature rankings are built based on the MV scores of the features, and the RR policy selects features from the top-most elements of the rankings.

3 Experiments

3.1 Experimental setting

The dataset we use in this work is a set of 15,763 hotel reviews we have obtained by crawling from the TripAdvisor Web site all the reviews related to hotels in the towns of XXXX and YYYY[6] (approximately 26,000 such reviews were obtained), and then applying a language recognition system, that we have implemented along the lines of [14], in order to filter out all reviews not in English[7]. Each review has a score of one to five "stars", both globally and for each of seven facets: "BusinessService", "CheckIn/FrontDesk", "Cleanliness", "Location", "Rooms", "Service", "Value". Aside from the "global" dataset, we have also defined seven facet-specific datasets, which contain all and only the reviews for which a label has been attributed for the given facet (not all reviews contain scores for all of the facets); the largest facet-specific dataset is "Value", with 12,038 reviews, while the smallest is "BusinessService" dataset, with 4,148 reviews. The label distribution is highly skewed, since 45% of all the reviews have a global score of 5 stars, 34.5% a global score of 4 stars, 9.4% 3 stars, 7.2% 2 stars and only 3.9% 1 star (the skew is even higher in the facet-specific datasets). This tends to make the system's task for the least frequent scores difficult. We have independently and randomly split each of the 8 datasets into a training set, containing 75% of the reviews of the entire dataset, and a test set, consisting of the other 25%[8].

[6] XXXX and YYYY reviews were crawled on May 12 and 14, 2008, respectively. Town names blotted out to preserve anonymity.

[7] Our implementation of this language recognition system is freely available for download from `http://` (URL removed to preserve anonymity).

[8] All the datasets discussed in this paper are available for download from `http://` (URL removed to preserve anonymity).

	Global		Average	
	MAE^{μ}	MAE^{M}	MAE^{μ}	MAE^{M}
MajorityLabel	0.657	1.896	0.773	1.600
BoW	0.621	0.799	0.803	1.160

Table 1. Baseline results. Lower values indicate better accuracy. "Global" stands for results on the global dataset; "Average" stands for average results across the seven facet-specific datasets.

Conforming to standard practice, as an evaluation measure we use *mean absolute error*, defined in terms of average deviation between the predicted and the true label. We report results using both a micro- and a macro-averaged version of MAE (respectively noted MAE^{μ} and MAE^{M}), and defined as

$$MAE^{\mu}(\hat{\Phi}, Te) = \frac{1}{|Te|} \sum_{x_j \in Te} |\hat{\Phi}(x_j) - \Phi(x_j)| \tag{1}$$

$$MAE^{M}(\hat{\Phi}, Te) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|Te_i|} \sum_{x_j \in Te_i} |\hat{\Phi}(x_j) - \Phi(x_j)| \tag{2}$$

where Te denotes the test set and Te_i denotes the set of test documents whose true label is y_i. In MAE^{μ} all examples count the same (since MAE^{μ} is computed by taking the deviation between predicted and true label for each document and then averaging across documents), while in MAE^{M} all labels count the same (since MAE^{M} independently computes average deviation for all test documents with a given label and then averages across labels). Which evaluation measure is the "right" one obviously depends on the application constraints.

3.2 Results and discussion

For POS-tagging the text of the reviews we have used the POS-tagging utility provided by the Natural Language Toolkit[9] (NLTK) package.

We provide two baselines, a "trivial" one ("MajorityLabel") in which all test documents are assigned the label most frequent in the training set, and a less trivial one ("BoW") based on ϵ-SVR and a simple bag-of-word representation with no feature selection. Table 1 reports MAE^{μ} and MAE^{M} values for the two baselines. An effectiveness value is provided for the global dataset in the left-hand side of the table; for the seven facet-specific datasets, an effectiveness value that averages across them (with each dataset counting the same) is provided in the right-hand side. Table 2 shows MAE^{μ} and MAE^{M} values obtained for various combinations of text representation method and feature selection method. In all experiments, the 10% top-scoring features are selected via the indicated feature selection method.

Several observations can be made based on these tables. The first is that representations more sophisticated than bag-of-words always provide superior

[9] `http://nltk.sourceforge.net`

	Global				Average			
	MAE^{μ}		MAE^{M}		MAE^{μ}		MAE^{M}	
	MV	**RRMV**	**MV**	**RRMV**	**MV**	**RRMV**	**MV**	**RRMV**
BoW	0.682	0.654	1.141	0.970	0.847	0.872	1.291	1.269
BoW+Expr	0.456	0.547	0.830	**0.657**	0.752	0.743	1.561	1.093
BoW+Expr+sGI	0.448	0.776	1.165	0.937	0.781	0.824	**1.008**	1.181
BoW+Expr+sGI+eGI	**0.437**	0.565	0.942	0.677	**0.733**	0.741	1.032	1.092

Table 2. Results obtained for various combinations of features and feature selection methods, with only 10% of the total number of features retained. "BoW" stands for bag-of-words, "Expr" for the expressions of Section 2.1, "sGI" and "eGI" for simple and enriched GI expressions, respectively. The best performing combinations are shown in **boldface**.

or much superior performance than BoW; **BoW+Expr+sGI+eGI** provides the best representation in 2 cases out of 4 (given by 2 evaluation measures $\times$ 2 feature selection methods), provides consistently good performance across the table, and provides very substantial improvements over pure bag-of words. For instance, in the "Global" experiments MAE^{μ} improves from .621 to .437 (a 29.6% relative improvement) over BoW, while MAE^{M} improves from .799 to .677 (a 15.3% relative improvement).

The second observation is that, as a feature selection method, MV generally outperforms RRMV on MAE^{μ}, but the contrary often happens on MAE^{M}. This can be explained by the fact that only RRMV places equal importance on all labels, by selecting some highly discriminating features for each label; as a consequence, RRMV tends to excel when the results are evaluated with a measure, such as MAE^{M}, that places equal importance on each label. Conversely, it is likely that for frequent labels MV finds many discriminating features, while it finds few for less frequent labels; as a consequence, MV tends to excel when the results are evaluated with a measure, such as MAE^{μ}, that in fact attributes more importance to more frequent labels. However, we should observe that retaining only 10% of the total amount of features has proven a suboptimal choice, as can be observed by the general deterioration in performance that resulted in moving from BoW with all features (2nd line of Table 1) to BoW with 10% of the features only (1st line of Table 2). In the future we plan to experiment with different, less aggressive levels of feature selection.

The third observation is that, when MAE^{ν} is used, in the "Global" experiments the "trivial" baseline (MajorityLabel) is only marginally improved upon by the BoW baseline (a non-trivial baseline in which a sophisticated learning device such as ϵ-SVR is involved), and even outperforms it on the "Average" experiments! This can be explained by the fact that the distribution of labels in these datasets is highly skewed towards a majority label (as noted in Section 3.1, this is especially true in the facet-specific datasets), with the consequence that the trivial classifier that assigns all test objects to the majority label may be hard to beat by any non-trivial classifier. In the light of this, the improvements obtained over BoW thanks to our methods acquire even more value.

4 Related work

In this section we review related work on the analysis and rating of product reviews, focusing on the differences between these approaches and ours.

The work of Dave et al. [15] is the first to address the problem of scoring product reviews based on an analysis of their textual content. Unlike us, they address binary classification, only distinguishing between Positive and Negative reviews. Based on a corpus of reviews that they crawled from the Web they design and test a number of methods for building product review binary classifiers.

Unlike [15], [17] addresses product review scoring with respect to an ordinal scale of more than 2 values. Unlike us, their work is focused on the learning approach to be used. They propose and compare a multi-class SVM classifier, ϵ-SVR, and a meta-algorithm based on a metric labeling formulation of the problem. A related work is [16], where a semisupervised algorithm is applied that learns to rate product reviews from both rated and unrated training reviews. Also devoted to testing learning algorithms for rating product reviews is [20], which addresses multi-facet review rating on a corpus of Japanese reviews.

In [19] rating inference is addressed in a simplified way: while the reviews in the training set are labeled according to a five-point scale, the system described is only capable of assigning labels in the set {Positive, Neutral, Negative}, thus "compressing" the original rating scale to a coarser one. This is very different from what we do, since our system is capable of predicting labels on ordinal scales containing an arbitrary number of labels.

In [22] a new task in product review analysis is identified, i.e., the prediction of the utility of product reviews, which is orthogonal to scoring by perceived quality. The authors formalize the problem in terms of linear regression and experiment with two types of regression algorithms, ϵ-SVR and *simple linear regression* (SLR) as implemented in `WEKA`.

In [18] online hotel reviews are ranked in a way similar to ours. The authors manually build a lexicon of expressions conveying either positive or negative sentiment with respect to the domain of hotel reviews. However, their experimental evaluation is weak, since a very small test set of reviews (about 250) is used, and the evaluation simply consists in ranking pairs of reviews according to which is more positive than the other.

5 Conclusions

We have presented a system for automatically rating product reviews that independently rates many distinct aspects ("facets") of the product, so that the same review could be given different ratings for different facets. We have investigated various methods for the generation of the vectorial representations of the reviews to be fed to the learning system, including methods for the generation of complex features based on the detection of part-of-speech patterns, methods for enhancing the statistical robustness of these patterns through the application of a lexicon of opinion-carrying words, and feature selection methods for ordinal regression. These latter methods, in particular, had never been presented in

the literature, and are original contributions of this work. We have shown that a combination of all these methods substantively outperforms a baseline consisting of a bag-of-words representation.

Rating product reviews is a fairly recent application, so a lot of research still needs to be done. In the future, we would like to work on several problems that this work has highlighted, the first of which has to do with creating a larger and more varied dataset that can be considered representative of the many types of reviews one encounters for a given type of product. We intend to crawl a much larger reviews dataset, representative of the many types of destination which hotels cater for. The current dataset only represents towns interesting for their works of art, but other types of destination should be represented such as, e.g., seaside resorts, mountain destinations, and the like. The reason why such variety may be desirable is that different language may be used to praise a hotel in a seaside location than a hotel in a business-oriented town. Another aspect we would like to work on is multilinguality: currently, non-English reviews have been excluded from the dataset, but in the future they might be used to form a multilingual corpus, partitioned into as many sub-corpora as the languages represented inside it. For instance, this might lead to the use of techniques from cross-language information retrieval in order to profit from the existence of reviews of the same hotel in more than one language.

TripAdvisor reviews also contain some extra information that we have not used, such as for which type of customers the hotel can be recommended and for which other types it cannot. This is a dimension that we have not investigated; in the future, techniques similar to the ones we have used here might be used in order to assess, again on an ordered scale, how much a given hotel may be recommended for a particular type of users.

References

1. Gretzel, U., Yoo, K.Y.: Use and impact of online travel review. In: Proceedings of the 2008 International Conference on Information and Communication Technologies in Tourism, Innsbruck, AT (2008) 35–46
2. Jindal, N., Liu, B.: Review spam detection. In: Proceedings of the 16th International Conference on the World Wide Web (WWW'07), Banff, CA (2007) 1189–1190
3. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval **2**(1/2) (2008) 1–135
4. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Computation **12**(5) (2000) 1207—1245
5. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at `http://www.csie.ntu.edu.tw/ cjlin/libsvm`.
6. Caropreso, M.F., Matwin, S., Sebastiani, F.: A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In Chin, A.G., ed.: Text Databases and Document Management: Theory and Practice. Idea Group Publishing, Hershey, US (2001) 78–102
7. Stone, P.J., Dunphy, D.C., Smith, M.S., Ogilvie, D.M.: The General Inquirer: A Computer Approach to Content Analysis. The MIT Press, Cambridge, US (1966)

8. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Proceedings of the 11th International Conference on Machine Learning (ICML'94), New Brunswick, US (1994) 121–129
9. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning (ICML'97), Nashville, US (1997) 412–420
10. Miller, A.: Subset selection in regression. Second edn. Chapman and Hall, London, UK (2002)
11. Geng, X., Liu, T.Y., Qin, T., Li, H.: Feature selection for ranking. In: Proceedings of the 30th ACM International Conference on Research and Development in Information Retrieval (SIGIR'07), Amsterdam, NL (2007) 407–414
12. Mukras, R., Wiratunga, N., Lothian, R., Chakraborti, S., Harper, D.: Information gain feature selection for ordinal text classification using probability redistribution. In: Proceedings of the IJCAI'07 Workshop on Text Mining and Link Analysis, Hyderabad, IN (2007)
13. Forman, G.: A pitfall and solution in multi-class feature selection for text classification. In: Proceedings of the 21st International Conference on Machine Learning (ICML'04), Banff, CA (2004) 38–45
14. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94), Las Vegas, US (1994) 161–175
15. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th International Conference on the World Wide Web (WWW'03), Budapest, HU (2003) 519–528
16. Goldberg, A.B., Zhu, X.: Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In: Proceedings of the HLT/NAACL Workshop on Graph-based Algorithms for Natural Language Processing, New York, US (2006)
17. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, US (2005) 115–124
18. Pekar, V., Ou, S.: Discovery of subjective evaluations of product features in hotel reviews. Journal of Vacation Marketing **14**(2) (2008) 145–156
19. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP'05), Vancouver, CA (2005) 339–346
20. Shimada, K., Endo, T.: Seeing several stars: A rating inference task for a document containing several evaluation criteria. In: Proceedings of the 12th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'08), Osaka, JP (2008) 1006–1014
21. Snyder, B., Barzilay, R.: Multiple aspect ranking using the good grief algorithm. In: Proceedings of the Joint Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technology Conference (NAACL/HLT'07), Rochester, US (2007) 300–307
22. Zhang, Z., Varadarajan, B.: Utility scoring of product reviews. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM'06), Arlington, US (2006) 51–57