Hotel recommendation based on reviews opinion mining

Xinyu Yao, Xiaotian Huang, Zhilin Wang, Chao Zhou

1. Introduction

While reading reviews of hotels, there is specific information we are looking for: amenities the hotel provides, such as pool, parking, etc, if the location of the hotel is convenient, and if the room is clean, quiet, and warm. Here using hotel reviews dataset provided by Datafiniti's Business Database[1], we provided hotel comparison and recommendation based on reviews opinion mining. This dataset has 10000 reviews for 1000 hotels, with corresponding hotel location, name, and rating. Therefore, we could provide hotel comparison and recommendation based on sentiment analysis of hotel reviews and use the rating provided by the reviewer to evaluate the performance of sentiment analysis.

2. Background

TripAdvisor Annotated Dataset[2] has annotations related to nine aspects typically mentioned in hotel reviews: "rooms", "cleanliness", "value", "service", "location", "checkin", "business", "food", "building" and two "catch-all" aspects "Other" and "NotRelated", for a total of 11 aspects. Using the annotated dataset, a multi-label classification model was trained to classify sentences into 11 aspects.

Term Frequency Inverse Document Frequency (TF-IDF)[3] calculates values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in. sklearn.feature_extraction.text.TfidfVectorizer was used to convert reviews to a matrix of TF-IDF features for training classification models.

For sentiment analysis, the sentiment property of the textblob[4] was used, which returns polarity, a float within the range [-1.0, 1.0]. SentiWordNet[5], a lexical resource for opinion mining, was also used, which assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity.

Part-of-speech (POS) tags assigned by spaCy's models[6] were used to assign sentiment scores to terms mentioned in reviews. POS explains how a word is used in a sentence. There are eight main parts of speech - nouns, pronouns, adjectives, verbs, adverbs, prepositions, conjunctions, and interjections.

[1] Hotel Reviews https://www.kaggle.com/datafiniti/hotel-reviews

[2] Marcheggiani, D., Täckström, O., Esuli, A, Sebastiani, F.: Hierarchical Multi-Label Conditional Random Fields for Aspect-Oriented Opinion Mining. In: Proceedings of the 36th European Conference on Information Retrieval (ECIR 2014).

[3] Ramos, J., 2003, December. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (Vol. 242, pp. 133-142).

[4] Textblob https://textblob.readthedocs.io/en/dev/quickstart.html

[5] Baccianella, S., Esuli, A. and Sebastiani, F., 2010, May. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec* (Vol. 10, No. 2010, pp. 2200-2204).

[6] Part-of-speech tagging https://spacy.io/api/annotation#pos-tagging

3. Scope

3.1 Hotel Comparison and Recommendation Based on Aspects Preference

A multi-label classification model was trained to classify sentences in reviews to aspects, then sentiment scores were assigned to corresponding aspects. In this way, we could provide hotel comparison and recommendation according to aspects preference selected by customers. Here are the steps of our attempt:

- TripAdvisor Annotated Dataset[2] was split into trains and test sets, then the classifier pipeline of TfidfVectorizer and LabelPowerset was trained to classify sentences into aspects. The best classifier was saved to use in the next step.
- Polarities, returned from the sentiment property of textblob[4], were assigned to corresponding aspects, specified by our best classifier. Feature 'polarity' was created in this step.
- Adjectives, having positive and/or negative scores according to SentiWordNet[5], were subtracted, then terms associated with the adjectives were found through POS[6], and were assigned the sentiment scores. 'Term_score_dict' generated in this step will be used to rank hotels on keywords.
- The aspects, most similar to the terms according to word2vec[7], were assigned the sentiment scores of the terms. Feature 'term_score' was created in this step.
- A linear regression model was fitted on features of 'polarity' and 'term_score' and target of 'review.rating' and used to generate ratings in aspects.

3.2 Ranked Hotel List on Keyword Stated by User

Hotels in users' aiming areas were ranked on scores of terms that are similar to the keyword stated by the user. Here are the steps to do so:

- Check the similarity between terms in 'term_score_dict', generated from reviews of hotels in users' aiming areas, and the keyword stated by the user. The similarity is returned from gensim.models.Word2Vec.n_similarity.
- Scores of terms similar to the keyword are summed up and divided by num_reviews to get an average keyword_score for each hotel.
- Return the sorted hotels list on keyword_score.

4. Outcome

4.1 Sentence classification into aspects

The best classifier we got is

```
Pipeline([('vect', TfidfVectorizer(max_df=0.5, ngram_range=(1, 2), use_idf=False, stop_words="english")),
          ('clf', LabelPowerset(LogisticRegression(solver='newton-cg', multi_class='multinomial', C=100)))])
```

which gets an f1_macro of 0.59 on the testset.

The confusion matrices are

```
Room:    [1184   24]  Cleanliness: [1263    2]      Value: [1218    8]  Business: [1070   54]
         [  45   19]               [   4    3]             [  28   18]            [ 117   31]
Service: [1199    2]     Location: [1123   32]    Checkin: [1096   27]      Food: [ 711  175]
         [  35   36]               [  45   72]             [  55   94]            [ 124  262]
Building:[ 990   62]        Other: [1003   59] Notrelated: [1175   22]
         [  59  161]               [  88  122]             [  45   30]
```

[7] Word2vec https://code.google.com/archive/p/word2vec/

The best classifier gets most sentences in ‘service’, ‘location’, ‘checkin’, ‘food’, ‘building’, and ‘other’ classified right. There are only 7 sentences labeled with the ‘cleanliness’ aspect, so more tagged sentences will improve the performance of classification.

4.2 Sentiment analysis

	reviews.text	aspects	reviews.rating	pred_rating	polarity	term_score
0	This hotel was nice and quiet. Did not know, there was train track near by. But it was only few train passed during our stay. Best Western changed hotel classification. The Plus category are not the same as before.	{OTHER, BUILDING, NOTRELATED}	3	4.173049	0.233333	0.625000
1	We stayed in the king suite with the separation between the bedroom and the living space. The sofa bed wasn't very good I had back discomfort by the day we left on our three night stay. The room is clean, and the king bed very comfortable. This hotel is located within walking distance to most places you will want to... More	{OTHER, CLEANLINESS, ROOMS, LOCATION}	4	4.219080	0.273889	0.343750
2	Parking was horrible, somebody ran into my rental car while staying there. I didn't get to try the breakfast, I was there for business so the restaurant opened to late for the business world to enjoy, I had to asked for coffee for my room, And the items in the vending machine were stale.	{OTHER, FOOD, BUSINESS, ROOMS}	3	1.106107	-0.450000	-0.375000
3	Not cheap but excellent location. Price is somewhat standard for not hacing reservations. But room was nice and clean. They offer good continental breakfast which is a plus and compensates. Front desk service and personnel where excellent. It is Carmel, no A/C in rooms but they have a fan for air circulation.	{SERVICE, ROOMS, FOOD, CLEANLINESS, LOCATION, VALUE}	5	4.893079	0.430556	0.500000
4	If you get the room that they advertised on the website and for what you paid, you may be lucky.If you stay many days , they will give you the not so good rooms.Nobody wants to stay in these rooms: low light/dark rooms, near pool, noisy, smelly bathrooms, or difficult access. If you stay one-two days you will get probably... More	{OTHER, VALUE, ROOMS}	2	3.788863	0.288333	-0.875000

Fig 1. DataFrame of ‘reviews.text’ and ‘reviews.rating’ from the raw dataset and ‘aspects’, ‘pred_rating’, ‘polarity’, and ‘term_score’ generated

A linear regression model was fitted on features of ‘polarity’ and ‘term_score’, and target of ‘reviews.rating’ and got Mean Square Error (MSE) of 0.58. With a good classifier to classify reviews into aspects and a precise linear regression model to predict ratings, we could then generate ratings in aspects from reviews.

4.3 Hotel comparison in aspects

Suppose users are interested in Best Western Plus Hacienda Hotel Old Town (BW) and Hampton Inn San Diego Del Mar (Hampton), which both have ‘reviews.rating’ of 4.1. If checkin, services, business, value, and food weight more in users’ decisions, we would recommend Hampton to stay. A side note here is that aspect of business may not be taken into consideration as much as other aspects, as though customers are not satisfied with the business performance of BW with a score of 2.3 given, it still gets a good overall rating.

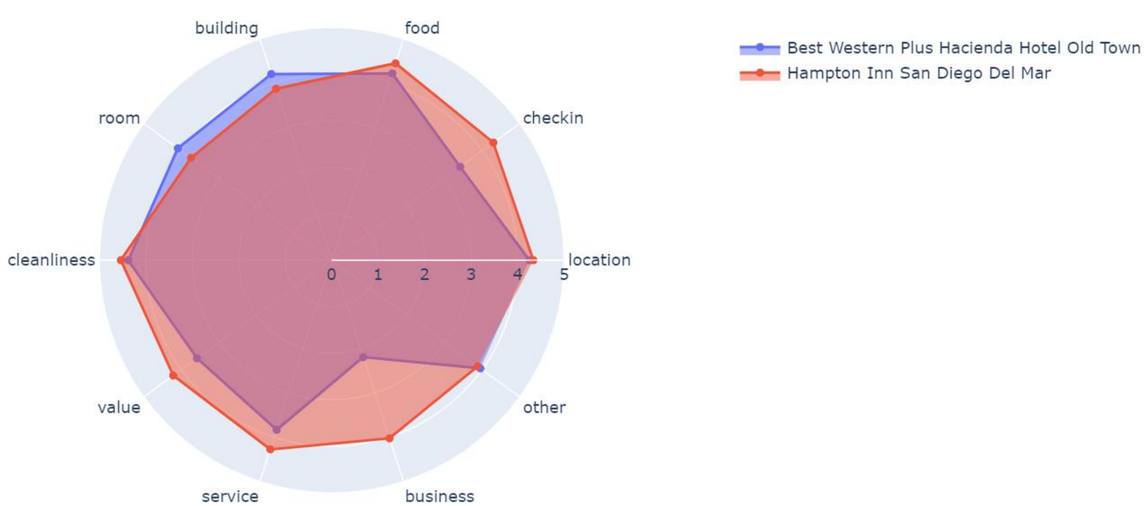

Fig 2. Comparison of Best Western Plus Hacienda Hotel Old Town and Hampton Inn San Diego Del Mar

4.4 Ranked hotel list on keyword stated by user

Suppose the user is looking for hotels in San Diego and is interested in breakfast quality of hotels. We would recommend Hampton Inn San Diego Del Mar. This recommendation agrees with the comparison result we got from the last section that Hampton Inn San Diego Del Mar has better food compared to Best Western Plus Hacienda Hotel Old Town.

```
rank_hotel = rank_hotel('breakfast')
```

```
nice breakfast
assorted bread
cereal
sausage
bread
breakfast choice
continental breakfast
meal
food
adequate breakfast
```

```
sorted(rank_hotel.items(), key=lambda v: v[1], reverse=True)
```

```
('Hampton Inn San Diego Del Mar', 0.5911387310041553),
('Best Western Plus Bayside Inn', 0.5585874690118137),
('Best Western Yacht Harbor Hotel', 0.5267629840663003),
('Best Western Plus Hacienda Hotel Old Town', 0.5208325375373
('Best Western Mission Bay', 0.486439825948136),
('Best Western San Diego/Miramar Hotel', 0.414747340446431),
('The Pearl Hotel', 0.41018644261617576),
('Quality Suites San Diego SeaWorld Area', 0.3345512124113481
('Best Western Seven Seas', 0.3005518661088366),
('Ocean Park Inn', 0.1302250171813301)]
```

Fig 3. Part of terms subtracted from reviews that are similar to the keyword ‘breakfast’ (left), ranked hotel list on the keyword ‘breakfast’(right)

More details could be found on github. https://github.com/xinyuyao22/Topics-Extraction-Hotel-Reviews

5. Challenges

a. We tried word2vec, using the pre-trained Google News corpus word vector model, which transform reviews into 300-dimensional vectors, however, this does not work well, probably because the word vector model is trained on news corpus and there is not much variance among hotel reviews on these 300 dimensions.

b. To get sentiment scores for aspects, we first classify sentences into aspects and then assign sentiment polarity to aspects. However, opposite sentiments towards aspects in one sentence may be missed, such as ‘The location is convenient, but the bed is not very good’. The polarity returned is -0.27, though the sentiment for location is positive. So, for compensation, we assign the scores of terms similar to specific aspects and use these two features to fit a linear regression model.

6. Future Work

a. To train a better classifier, we could: 1) tag more sentences in aspects of ‘value’, ‘cleanliness’, etc; 2) recognize and replace pronouns using Neural Coref v2.0; 3) using textblob to do spell correction before analysis; 4) use pre-trained word2vec specified for review text; 5) model ensemble several classifiers.

b. The way we used to generate term_dict is to catch adjective words that have positive or negative sentiments and then assign the sentiment score to the noun described. However, for sentences like ‘I hate the food here’, we cannot get the sentiment. We could study more on different sentence patterns and structures.

c. Up to this point, we could take only one keyword as input from the user and return a ranked hotel list. The most general interface would be a single text field that would allow users to express preferences using natural keywords. Aspects of the query can then be obtained using various query segmentation techniques. It would be more convenient if allowing users to incrementally add preferences as needed instead of re-entering the entire query.