Summarizing Opinion from Customer Reviews

Created by Lance Deng on Jan 2, 2018

1. Problem Definition (Problem Selection)

The given dataset contains customer reviews in Chinese and English for diapers of different brands. Some given questions include: We are interested to learn what consumers are telling us about our products. What do they like and dislike? What actions might we consider based on this feedback?

I explored the dataset by checking the distributions of variables and sample some review texts[1]. To quickly get some idea of what the customers are talking about, I applied a very basic version of topic model on the reviews[2]. Based on the results, I can see that the reviews contain topics such as product quality, whether a product is genuine or fake, marketplace, logistics, brand perception, etc. However, the topic model does a poor job in grouping opinions by similar aspects.

Based on the given questions and the given dataset, I would like to propose an opinion summary task. An opinion object may contain the name of an entity, an aspect of the entity, the sentiment on the aspect, the opinion holder, and the time (Liu 2012). Due to the time limit, I define an opinion object as a simple pair, *(aspect, opinion)*. Some examples are *trustworthy brand*, *fast shipping*, and *pungent smell*. The aspect refers to a certain feature or aspect of a product. The opinion describes the aspect.

The input to this task is a large set of customer reviews. The output of this task are opinion pairs grouped by aspects. This allows decision makers to quickly summarize what customers like and dislike about the product.

2. Literature Review

As a scientific approach, I typically try to find state-of-the-art algorithms first, and then, think about how it can be improved. The aforementioned opinion summarizing task is commonly referred as aspect-based sentiment analysis in opinion mining literature. I searched for papers related to aspect-based sentiment analysis in Google Scholar. Given the time limit, I only went through ten research papers. These papers used different approaches such as topic modeling, deep learning, and linguistic rules. The rule-based approach seems to be most feasible given the target turnaround time and the lack of annotated data. I identified two well received papers, Qiu et al. (2011) and Poria et al. (2014). Both papers used customer review data, which is similar to our problem domain.

Qiu et al. (2011) proposed an algorithm to extract aspects and opinion words from customer reviews. It takes a lexicon expansion approach. This approach identifies new aspects and opinion words based on word associations and a list of seed words. The paper also

---

[1] See “S0_Exploration.ipynb”
[2] See “topic modeling.ipynb”

proposed rules for associating words. These rules are based on dependency parsing and part-of-speech (POS) tagging. The rules are summarized as follows[3]:

For aspects:

- R1-1: A noun modified by a known adjective. (e.g., a good screen)
- R1-2: Two nouns having a *nsubj* relationship. (e.g., iPod is the best mp3 player.)
- R3-1: Two nouns having a *conj* relationship. (e.g., play DVD with audio and video.)
- R3-2: Two nouns having *nsubj* and *dobj* relationship with the same word. (This camera has a great lens.)

For opinion words:

- R2-1: Similar to R1-1 with the aspect known.
- R2-2: Similar to R1-2 with the aspect known.
- R4-1: Two adjectives having a *conj* relationship.
- R4-2: Two adjectives modifying the same noun.

Poria et al. (2014) proposed rules for identifying aspects. The study claimed that their rules outperformed that in Qiu et al. (2011).

- A noun modified by an adjective or an adverb.
- A noun in a *dobj* relationship.
- A noun and its verb if the verb is modified by an adjective or an adverb.
- Two nouns in a *nmod* relationship.
- Two nouns in a *compound* relationship.
- A noun in a *nsubj* relationship with an adjective.
- A verb in a *xcomp* with an adjective.

These papers do not have the algorithms that directly generate the opinion pairs I previously defined. Neither is it feasible to implement these algorithms give the time constraint. However, based on these two papers, I think it is a good starting point to manually code some rules based on dependency types and POS tags.

3. Approach

3.1 Extracting Opinion Pairs

I have more experience analyzing English texts. But some of the English translation from Chinese are not very accurate. As a result, I explored a few tools for applying dependency parsing and POS tagging on both Chinese and English texts[4]. After some qualitative evaluation, I chose to work on English texts using Stanford Parser. The parser does a good job in collapsing certain indirect dependencies, which is convenient for generating linguistic rules.

[3] See the following webpage for an explanation of dependency types: http://universaldependencies.org/docsv1/en/dep/index.html

[4] See “Chinese NLP.ipynb” and “English NLP.ipynb”.

Based on the distribution of the dependencies and POS tags, I started to create linguistic rules that can be used to extract opinion pairs from English reviews[5]. These rules are:

- amod(N, A): a noun modified by an adjective.
- nsubj(A, N): a noun as the subject of an adjective.
- advmod(N, R): a noun modified of an adverb.

These patterns cannot capture all types of opinion pairs. But they do account for a large number of them. After applying these patterns, I extracted 80,942 potential opinion pairs. Among them, there are 2,261 potential aspects. Many of these aspects refer to the same thing. Thus, the next step is to group similar aspects.

3.2 Grouping Similar Aspects[6]

In order to measure similarity between aspects, we need a feature vector for each aspect. I used pretrained Word2vec embeddings as feature vectors. That is, each aspect is mapped to a 300-dimension continuous vector using Word2vec. Subsequently, a clustering algorithm is used to group similar aspects. In order to get smaller groups, I set the number of cluster to 500. Some clustering results are shown below:

- Cluster 56: price, prices, cost, expense, spending, fee, costs
- Cluster 368: odor, smell, scent, odors
- Cluster 0: brand, brands, product, products, promotion, branding, marketing, promotions

3.3 Sentiment Classification

In order to extract both positive and negative opinion, I used StanfordNLP to classify the sentiment of each opinion pair as positive, negative, and neutral.

3.4 Opinion Summary

I chose some representative aspects, such as: "price", "absorption", "odor", "package", "brand", "shipping", "elasticity", "padding", "lining", "material". For each of these aspect, I retrieved opinion pairs having an aspect that is in the same cluster. Then, I grouped these opinion pairs by sentiment and brand. Up to 10 most frequent opinion pairs are then displayed. Some of the results are shown below.

```
Query word: price      Word cluster: 56
Aspects: ['price' 'prices' 'cost' 'expense' 'spending' 'fee' 'costs']

*Positive*----------------------------------------
brand             Huggies  Merries  Pampers  Total
pair
affordable price       92       51       71    214
good price             64       66       73    203
cheaper price          59       51       54    164
expensive price        58       47       48    153
high price             61       31       55    147

*Neutral*----------------------------------------
```

[5] See “S1_extract.ipynb”
[6] See S2_consolidate.ipynb

brand pair	Huggies	Merries	Pampers	Total
high cost	4	5	7	16
better price	5	4	2	11
affordable prices	2	3	0	5
high prices	3	2	0	5
curious price	3	0	0	3

Negative--

brand pair	Huggies	Merries	Pampers	Total
cheap price	251	142	261	654
low price	21	16	22	59
bad price	10	5	10	25
cheap prices	0	1	6	7
worse price	5	0	0	5

==

Query word: odor   Word cluster: 368
Aspects: ['odor' 'smell' 'scent' 'odors']

Positive--

brand pair	Huggies	Merries	Pampers	Total
strong smell	1	4	8	13
good smell	3	4	4	11
great smell	0	2	3	5
strong odor	1	0	2	3
genuine smell	2	0	0	2

Neutral--

brand pair	Huggies	Merries	Pampers	Total
pungent smell	13	29	41	83
big smell	5	7	32	44
strange smell	4	11	11	26
little smell	5	5	10	20
heavy smell	3	5	11	19

Negative--

brand pair	Huggies	Merries	Pampers	Total
bad smell	3	3	12	18
unpleasant smell	3	3	5	11
cheap smell	0	0	1	1
obvious odor	0	0	1	1
so-called smell	0	0	1	1

==

Query word: package   Word cluster: 253
Aspects: ['packs' 'package' 'packages' 'pack' 'kit' 'bundle']

Positive--

brand pair	Huggies	Merries	Pampers	Total
good package	17	21	18	56
new package	17	10	12	39
big package	9	9	13	31

```
large package           5        4        9     18
complete package        5        5        2     12

*Neutral*---------------------------------------
brand           Huggies  Merries  Pampers  Total
pair
small package        17        9       18     44
few packs             6       13        4     23
intact package        1        3        6     10
first package         3        5        2     10
second pack           4        6        0     10

*Negative*---------------------------------------
brand           Huggies  Merries  Pampers  Total
pair
dirty package         1        3        3      7
rotten package        3        4        0      7
fake package          0        6        0      6
thin package          3        2        0      5
bad package           1        2        0      3
```

Some opinion pairs do not exhibit a large variation across different brands. But others do. In the examples above, we can observe that Pampers has more complaints about its odor comparing to other brands.

4. Evaluation

The overall effectiveness of this approach can be quantitatively evaluated upon the availability of annotated data. Evaluation metrics may include accuracy, precision, recall, and F-measure.

5. Conclusion

In this project, I focused on an opinion summary task. Given English customer reviews, I proposed an approach to generate opinion pairs. I used machine learning and NLP technique such as dependency parsing, topic modeling, word embeddings, sentiment analysis, and clustering. A qualitative evaluation suggests that this approach is promising but also need to be further improved.

6. Limitations and Future Work

Due to the time constraint, there are many limitations of the proposed approach, which can be further improved given more resources.

The first limitation is the limited number of rules I used. More rules can be coded to improve both the prevision and recall of opinion pairs. If I were given a two-week timeframe, I would try to implement the algorithms in Qiu et al. (2011) and Poria et al. (2014).

The second limitation is that the proposed approach does not distinguish between explicit and implicit aspects. For example, “price” is an explicit aspect while “expensive” implicitly refers to “price”. Future work may pay more attention to implicit aspects.

The third limitation is that the NLP tools (e.g., parser, word embeddings, sentiment analysis) were not fine-tuned for our dataset. Future effort can also focus on optimizing these tools for our problem domain.

The proposed approach involves a number of hyper parameters. Future work can also optimize results by tuning hyper parameters.

Some of the translations from Chinese to English are not accurate. Future work can focus on directly analyzing the Chinese texts.

The proposed approach is rule-based. A number of unsupervised learning methods, such as variations of topic modeling and representation learning, should be attempted in future work.

References:

Liu, B. 2012. "Sentiment Analysis and Opinion Mining," *Synthesis Lectures on Human Language Technologies* (5:1), pp. 1-167.

Poria, S., Cambria, E., Ku, L.-W., Gui, C., and Gelbukh, A. 2014. "A Rule-Based Approach to Aspect Extraction from Product Reviews," *Proceedings of the second workshop on natural language processing for social media (SocialNLP)*, pp. 28-37.

Qiu, G., Liu, B., Bu, J., and Chen, C. 2011. "Opinion Word Expansion and Target Extraction through Double Propagation," *Computational linguistics* (37:1), pp. 9-27.