Aspect Based Sentiment Analysis

Akshita Jha, Shubham Ghiya and Juhi Ghosh
Team 34, IIIT Hyderabad

***Abstract*— We are concerned with the task of Aspect Based Sentiment Analysis (ABSA), where the goal is to identify the aspects of given target entities and the sentiment expressed towards each aspect in restaurant reviews. Also, we identify the categories of the aspect term and the category polarity for each review. Our approach is rule-based and we have come up with various heuristics to solve this non-trivial problem. When evaluated using SemEval-2014 dataset, we obtain decent results for aspect based sentiment analysis.**

***Keywords* - Sentiment Analysis, Aspect, Category Detection, Wordnet, Natural Language Processing**

I. INTRODUCTION

In recent years, the explosion of social networking sites, blogs and review sites provide a lot of information. Millions of people express uninhibited opinions about various product features and their nuances. This forms an active feedback which is of importance not only to the companies developing the products, but also to their rivals and several other potential customers. Sentiment Analysis is the task of tapping this goldmine of information. It retrieves opinions about certain products or features and classifies them as recommended or not recommended, that is positive or negative. The sentiment regarding a particular product in a review is seldom explicitly positive or negative; rather people tend to have a mixed opinion about various features, some positive and some negative. Thus the feature specific opinion matters more than the overall opinion. Consider a review: *The food was good but servicing was a bit slow*. This sentence has a mixed emotion. The emotion regarding food is positive whereas that regarding servicing life is negative. Hence, it is of utmost importance to extract only those opinions relevant to a particular feature (like *food* or *service*) and classify them, instead of taking the complete sentence and the overall sentiment. There are two types of aspects in aspect-based opinion mining: explicit and implicit. Explicit aspects are concepts that explicitly denote targets in the opinionated sentence. For instance, in the above example, *food* and *service* are explicit aspects as they are explicitly mentioned in the sentence. On the other hand, an aspect can also be expressed indirectly through an implicit aspect clue (IAC), e.g.,in the sentence *This Deli9 pizza is good and quite cheep*, which implicitly provides a positive opinion about the aspects pizza and price of the entity pizza. In this paper, we propose an approach to detect explicit aspects and implicit aspects from restaurant reviews. We also map these to their respective aspect categories. In this paper, we present a rule-based approach that exploits common-sense knowledge and sentence dependency trees to detect both implicit and explicit aspects. The paper is organized as follows: Section 2 presents the literature in aspect extraction; Section 3 describes in detail the aspect and category extraction approach; Section 4 describes the results of the experimental evaluation; and Section 6 concludes the paper.

II. RELATED WORK

Aspect extraction from opinionated text was first studied by Hu and Liu (Hu and Liu, 2004), who also introduced the distinction between explicit and implicit aspects. The authors only dealt with explicit aspects by adopting a set of rules based on statistical observations. Their algorithm detects whether a noun or noun phrase is a product feature or not by computing PMI between the noun phrase and the product class. Wu *et. al* use phrase dependency parsing for opinion mining. In dependency grammar, structure is determined by the relation between a head and its dependents.The dependent is a modifier or complement and the head plays a more important role in determining the behaviors of the pair. The authors want to compromise between the information loss of the word level dependency in dependency parsing as it does not explicitly provide local structures and syntactic categories of phrases and the information gain in extracting long distance relations. Hence they extend the dependency tree node with phrases.

Most of the works mentioned above require labeled datasets for training their models for each of the domains. The works do not exploit the fact that majority of the reviews have a lot of domain independent components. If those domain independent parameters are used to capture the associations between features and their associated opinion expressions, the models would capture majority of the feature specific sentiments with minimal data requirement.

III. METHOD

To evaluate explicit aspect extraction algorithm, we used the Semeval 2014 dataset[1] . The proposed method divides the task into 4 subtasks: (*a*) Preproccesing (*b*) Aspect term extraction (*c*) Aspect term Polarity (*d*) Aspect Category Detection (*e*) Aspect Category Polarity

A. Pre-Processing

Pre-processing is key for aspect parsing. The pre-processsing module of the proposed framework consists of two major steps: firstly we extract the nouns and adjectives in a sentence using *Nltk POS Tagger* and secondly the sentence

[1] http://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools

dependency tree is obtained through *Stanford Dependency-Parser*[2].

B. Aspect Term Extraction

Aspect term extraction is the key part of the project. Quality of aspect terms detected will result in performances of other subtasks. Words that follow following rules in *Stanford Dependency Parser* are filtered as aspect terms:

- If a word *t* has any adjectival complement *advmod* relationship with *h* than *h* is extracted as an aspect term.
- If a word *t* has an agent *agent* relationship with a word *h* than *h* is extracted as an aspect term.
- If a word *t* has an adjectival modifier *amod* relationship with a word *h* than *h* is extracted as an aspect term.
- If a word *t* has an direct object *dobj* relationship with a word *h* than *h* is extracted as an aspect term.
- If a word *t* has an nomial subject *nsubj* relationship with a word *h* than *h* is extracted as an aspect term.
- If a word *t* has an passive nominal subject *nusbjpass* relationship with a word *h* than *h* is extracted as an aspect term.
- If a word *t* is an object of preposition *pobj* relationship with a word *h* than *h* is extracted as an aspect term.
- If a word *t* has an open clausal complement *xcomp* relationship with a word *h* than *h* is extracted as an aspect term.

Additional Rules :

- In addition to these rules the words which follow the conjunction relationship *conj* with the extracted aspect words are also extracted as aspect terms.
- After extracting all the terms , the terms which are not nouns are removed from the aspect term list.

To extract the terms defining quality of these aspect terms following approach is implemented:

- Create a list of all adjective term in the sentence.
- If there are no adjectives in a sentence than extract all the verbs in sentence after removing stop words in a list.
- All words which are related to the aspect term upto a depth of 2 in *Stanford Dependency Relation* and present in the list are extracted as quality defining terms of aspect term.

C. Aspect term Polarity

For each aspect term we determined the terms defining the quality of the aspect terms with the help of Stanford Dependency Parser. The proposed framework determines the quality of each aspect term by calculating the sentiment belonging to the particular aspect term. This sentiment is calculated using the *TextBlob* python library. We then take a summation of the sentiments of the defining qualities for each aspect term. If the sentiment score of a term is greater than zero than it is extracted as *positive* aspect , if zero then a *neutral* aspect, otherwise a *negative* aspect term.

[2]http://nlp.stanford.edu:8080/parser

D. Aspect Category Detection

Category detection is one of the non-trivial tasks for this project. The possible values of the category field are: *food, service, price, ambience, anecdotes/miscellaneous*. The review can belong to one or more above of the categories. For category detection, we used the corpus developed by Cruz-Garcia[3] where IAC's are indicated and manually labeled by their corresponding aspect categories.

For our task, we extracted the sentences having implicit aspects and then extracted IACs for each of them, along with their corresponding labeled categories. For each IAC under every aspect category, synonyms and antonyms were obtained from WordNet (Fellbaum, 1998) and stored under the same aspect category. For example, *expensive* and its antonym *inexpensive* both have the same category *price*. Thus, a lexicon of 3398 IACs categorized into the above categories was built.

After, extracting the aspect term and it's describing words from a review, the describing words were looked up in the lexicon created. Depending on the category of the describing word, the aspect term was categorized. The similarity of the aspect term with the aspect categories was also taken into consideration. If the similarity score between an aspect term and a particular category was ¿= 0.5, then that aspect term was assumed to belong to that corresponding aspect category.

E. Aspect Category Polarity

To detect aspect category polarity in a review, all the aspect terms belonging to the particular category are clustered. After, this each aspect term's sentiment score (which was calculated for Subtask 2) is summed up and normalized. If the the sentiment score of a category is greater than zero than it's polarity is *positive* , if zero then it's of *neutral* polarity, otherwise it is *negative* polarity.

IV. Experimental Evaluation

SemEval'14 dataset consisted of 2000 restaurant reviews. Each sentence is tagged with a feature word, sentiment orientation of sentence with respect to the feature, category words and sentiment orientation towards each category. We calculated the accuracy of Aspect Term Extraction and Aspect Polarity, for our system for the below two scenarios:

1) When only **Nouns** are extracted as aspect terms.

   The precision, recall and F-Score for Aspect Term Detection has been shown in Table 1.

   - #System Aspect Terms=1995
   - #Gold Aspect Terms=737

Table 1: Aspect Term Detection

Precision	Recall	F-Score
0.266 (530/1995)	0.719 (530/737)	0.388

[3]Available from www.gelbukh.com/resources/implicit-aspect-extraction-corpus, April 9, 2016.

The details of Polarity Detection for this approach have been summarised in Table 2. The accuracy achieved via this approach is 0.3962264 (210/530).

Table 2: Aspect Polarity Detection

Label	Precision	Recall	F-measure
Negative	0.48(12/25)	0.106(12/113)	0.174
Neutral	0.232(86/370)	0.896(86/96)	0.3691
Positive	0.829(112/135)	0.366(112/306)	0.508
Conflict	NaN(0/0)	0(0/15)	NaN

2) Using the rules developed by our system, for the Aspect Term Extraction:

The precision, recall and F-Score for Aspect Term Detection has been shown for this approach in Table 3.

Table 3: Using Rules for Aspect Term Detection

Precision	Recall	F-Score
0.313 (346/1104)	0.469 (346/737)	0.376

The Aspect Polarity accuracy from this approach is 0.523 (181/346) (shown in Table 4)

Table 4: Aspect Polarity

Label	Precision	Recall	F-measure
negative	0.484(15/31)	0.203(15/74)	0.286
neutral	0.242(38/157)	0.691(38/55)	0.359
positive	0.810(128/158)	0.6184(128/207)	0.7014
conflict	NaN(0/0)	0(0/10)	NaN

The details of Aspect Category Detection are as follows (Table 5):

- #System Aspect Categories=1052
- #Gold Aspect Categories=752

Table 5: Aspect Category Detection

Precision	Recall	F-Score
0.423 (448/1052)	0.596 (448/752)	0.496

The details of Aspect Category Detection have been summarised in Table 6. The accuracy of our system is 0.511 (228/446)

Table 6: Category Polarity Extraction

Label	Precision	Recall	F-measure
Negative	0.435(20/46)	0.229(20/87)	0.301
Neutral	0.321(70/218)	0.722(70/97)	0.444
Positive	0.758(138/182)	0.575(138/240)	0.654
Conflict	NaN(0/0)	0(0/22)	NaN

V. CONCLUSION

In this project, we developed a system that extracts potential features from a review, detects their polarity, categorizes the aspect terms and detects the polarity of the category. We observed that the the aspect terms are not always Nouns. We also noticed that the errors in the output were not always due to our system, but because of errors in POS Tagging and building the Dependency Tree. Moreover, developing a rule-based system for review analysis does not give very good results.

The drawback of the system is that it cannot evaluate domain dependent implicit sentiment. Thus, the system will not be able to distinguish between *The story is unpredictable* (positive sentiment) and *The steering wheel is unpredictable* (negative sentiment). This is due to the usage of a generic sentiment lexicon, in the final stage, in rule-based classification. Supervised classification can help distinguish between these two sentiments but it needs tagged data and separate training for every domain. We also do not detect sarcasm and humour in our system.

Despite decent performance by our system, there is scope for imporvement. The future work can involve: discovering more rules for aspect term extraction, Combining existing rules for complex aspect extraction, making dictionaries more noise free, detecting conflicting sentiments and following a hybrid approach or supervised classification for better performance.

REFERENCES

[1] Christiane Fellbaum. 1998. WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press.

[2] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery Data Mining, pages 168177, Aug.

[3] Subhabrata Mukherjee, Pushpak Bhattacharyya,Feature Specific Sentiment Analysis for Product Reviews.

[4] Soujanya Poria, Erik Cambria,A Rule-Based Approach to Aspect Extraction from Product Reviews.

[5] Ankit Singh, Md. Enayat Ullah,Aspect based Sentiment Analysis

[6] Yuanbin Wu, Qi Zhang, Xuanjing Huang, Lide Wu,Phrase Dependency Parsing for Opinion Mining, EMNLP '09 Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2009, Volume 3