Detecting Opinion Sentences Specific to Product Features in Customer Reviews using Typed Dependency Relations

Ashequl Qadir
University of Wolverhampton
Stafford Street, Wolverhampton
West Midlands, WV1 1SB, UK
ashequl.qadir@wlv.ac.uk

Abstract

Customer reviews contain opinions of the customers who purchased products and expressed opinions concerning their satisfactions and criticisms. Due to vast availability of product reviews in the web, it is extremely time-consuming and at times confusing for a new customer to manually analyze the reviews prior to buying a product. Reviews generally involve the presence of product feature specific factual information along with the opinion sentences depicting the pros and cons of a bought product. The unstructured format of the text reviews from most of the web review sources necessitates the automatic identification of opinion sentences from the customer reviews, and also the identification of explicitly visible and implicitly present product features associated with the opinion sentences. In this paper, a process has been described where typed dependency relations such as open clausal complements or adjectival complements have been utilized to identify opinion sentences specific to product features. The typed dependency relations in the identified opinion sentences are then used to associate a product feature to an opinion sentence with the help of the product feature associated frequent words extracted from a previously managed customer review corpus.

Keywords

Product features, customer reviews, opinion sentences, typed dependency relations, frequent word association.

1.Introduction

After purchasing a product, customers quite often write their experiences in their reviews. These reviews contain their opinions about the product they purchased. These customer reviews are different from the traditional texts because they are written spontaneously and are small texts focused on a single topic or a product having several attributes and features. This relatively new type of texts mostly conveys sentiments about the topic or the purchased product and is getting widely popular day by day providing researchers with interests to explore a wide range of scopes and possibilities about how these texts can be processed and necessary information can be retrieved.

A new customer, before purchasing a product, quite often tends to look up the previously written reviews to analyze the positive and negative aspects of the product he intends to buy. This practice is increasing rapidly making it very important to formulate ways to process and retrieve information automatically from the text reviews. The products, for which the reviews are written, are associated with several product features, usually common to a particular product domain. The reviews can contain very general opinions such as '*I am very happy with this product*' or can also contain product feature specific opinions such as '*It is very easy and simple to use*', associated with a usability feature. Along with the opinion sentences, factual information such as '*it has a pink metal case*' can also be found in the reviews that do not contain any opinion of the reviewer; rather gives a factual description. As a result, before making a decision on the polarity of the opinions, it is very important to identify the opinion sentences and to identify the product features associated with them. Most of the popular products usually have many reviews written for them and it takes a significant amount of time to go through the review sentences manually in order to separate the opinion sentences from the others.

There are a number of review sources in the web where reviews can be found. E-commerce sites such as amazon, opinion sites such as epinions, forums, blogs etc are very well known sources for reviews and also very popular among the customers where reviews written by them can be found. Processing these mostly unstructured text reviews automatically is considered very challenging because of the frequent use of the informal expressions and terms, grammatically incorrect sentences, misspelled words etc. that can be occasionally found in the reviews.

Words forming a sentence have certain grammatical relations with each other based on their part-of-speech definitions, positions in the sentences etc. Some of these relations are representative of the functional features of a product for which the customers express their opinions. In this paper, a process has been described that utilizes the typed dependency relations of the words in sentences to identify opinion sentences written on product features. Because some of the relations are representative of the product features, these words are then utilized to assign a probable product feature to each of the opinion sentences under consideration. To utilize the dependency relations, Standord typed dependency relation representations [18] are chosen over PARC[20] representations because Standord typed dependency relations offers[17] more fine-grained distinctions in relations such as breaking down an unsubcategorized relation into several more distinctive relations like adjectival modifiers, prepositional relations, open clausal complements etc. This helps to obtain more precise dependency relations suitable for the designated purpose.

Events in Emerging Text Types (eETTs) - Borovets, Bulgaria, pages 38–43

2.Related Work

Opinion sentence identification has been mostly approached by the researchers by means of determining the presence of specific parts-of-speech such as adjectives, adverbs etc. or a list of seed words that may potentially represent opinions. Research of Wiebe[1] and Hatzivassiloglou et al.[2] showed that adjectives can potentially contribute towards identifying subjective sentences. Turney[3] used specific orientation of part-of-speech tags to extract phrases that can represent opinion. Godbole et al.[4], Kim et al. [5] used a small seed list of lexicons, expanded later, for their sentiment identification process. Riloff et al. [6] researched on identifying extraction patterns for subjective and objective sentences using subjective clues such as single words or N-grams. Wiebe er al.[7] worked on using word collocations that can act as subjectivity clues for identifying opinion sentences. Yu et al.[8] used the similarity between the opinion sentences within a given topic to identify opinion sentences and Naïve Bayes classification scheme to distinguish between opinion and factual sentences. Wilson et al.[9] used dependency relations of words as one of their syntactic clues for determining subjectivity strength. Fei et al[10] researched on utilizing the dependency relations of words in sentences for a target specific sentiment extraction.

Previous research works in product feature identification were mostly focused on explicit product features only. Yi et al.[11],[12] and Liu et al[13] worked on identifying explicit product features by extracting noun phrases of specific patterns. Popescu et al.[14] utilized parts and properties of a given product to identify product features. Ghani et al.[15] approached explicit product feature extraction as a classification problem. Qadir[16] used frequent word associations learned from a previously managed corpus to associate product features with sentences. Zhuang et al.[19] utilized dependency grammar graph to mine explicit feature-opinion pairs in movie review domain.

The approach described in this paper differs from the above mentioned previous researches by using Stanford typed dependency representations[17]. Specific typed dependency relations are utilized to differentiate opinion sentences from factual ones. Words forming the specific dependency relations are analyzed with frequent product feature associated words to assign a product feature to each of the opinion sentence.

3.Review Collection and Pre-processing

There are several product review sources available in the web. These sources can be e-commerce sites, opinion sites, forums, blogs etc. For this experiment, 100 reviews have been collected from amazon using amazon web services. Amazon web services (AWS) allows the developers to automatically collect plain text reviews. The collected reviews are from the domain '*Electronics*' and the product type is '*hard disk*'. 50 reviews have been used to identify the frequent words that are usually associated with the product features. This set of reviews has been used as a training corpus.

Each of the sentences in the set of reviews has been annotated manually with product feature titles. Sentences that do not convey any opinion of the reviewer have been tagged as 'No Opinion' and the sentences that convey only general opinions of the reviewers and not any product feature specific opinions are tagged as 'General'. Five other distinctive product feature titles have been identified from the reviews. Table 1 gives examples of the opinion lines that can be associated with these five different product features. These examples are taken from the collection of review texts.

Table 1. Product feature associated opinion sentences

Product Feature	Opinion Sentence
Usability	*'It was incredibly easy to set up and use.'*
Design	*'I like its design and the fact that I only need one cable.'*
Performance	*'Works perfectly and is completely reliable, no problem at all.'*
Portability	*'I found this product really useful for transport as it is that small.'*
Speed	*'The speed and capacity of the Passport drive are impressive.'*
General	*'A satisfying product.'*

The rest 50 reviews are kept for evaluating the process to identify opinion sentences and associate a product feature with each opinion sentence.

4.Methodology

The methodology section divides the whole process into two major tasks. To identify the opinion sentences, relevant typed dependency relations are selected and utilized. And to assign a product feature to each of the opinion sentences, frequently associated words are obtained from a previously managed corpus, normalized within the product feature scope by tf.idf metric and then utilized in the association process.

4.1Finding Opinion Sentences

4.1.1Typed Dependency Selection

Stanford Typed Dependencies Manual[18] gives definition to 55 binary grammatical relations between a governor and a dependent that can possibly be present in a sentence. From them, 3 of the relations have been selected as they can indicate a probable presence of product feature specific or general opinions in review sentences.

4.1.1.1acomp - Adjectival Complement

An adjectival complement (acomp)[18] of a VP is an adjectival phrase which functions as the complement (like an object of the verb); an adjectival complement of a clause is the adjectival complement of the VP which is the predicate of that clause. The governor component of the acomp typed dependency relation is a verb indicating a functionality of the product and the dependent component is an adjective indicating an opinion of the reviewer on that functionality. Table 2. gives examples of the components for acomp typed dependency relation taken from the review sentences of domain '*Electronics*'. Examples are given for the most frequent types of verb forms.

Table 2. Example of acomp relation as opinion indicator

Dependency Relation	Component Example	Indication
acomp	worked/VBD fine/JJ	Possible Opinion
acomp	proved/VBN reliable/JJ	Possible Opinion
acomp	works/VBZ well/JJ	Possible Opinion

4.1.1.2xcomp – Open Clausal Complement

An open clausal complement (xcomp)[18] of a VP or an ADJP is a clausal complement without its own subject, whose reference is determined by an external subject. In case of xcomp typed dependency relation, verb as the governor component and adjective as the dependent component and also adjective as the governor component and verb as the dependent component have been considered. Table 3. shows the examples taken from review lines in domain '*Electronics*' where xcomp can possibly indicate the present of an opinion in a review sentence.

Table 3. Example of xcomp relation as opinion indicator

Dependency Relation	Component Example	Indication
xcomp	easy/JJ use/VB	Possible Opinion
xcomp	rendering/VBG impossible/JJ	Possible Opinion
xcomp	found/VBD difficult/JJ	Possible Opinion
xcomp	makes/VBZ ideal/JJ	Possible Opinion
xcomp	find/VBP convenient/JJ	Possible Opinion
xcomp	experienced/VBN similar/JJ	Not Opinion

4.1.1.3advmod –Adverbial Modifier

An adverbial modifier(advmod)[18] of a word is a (non-clausal) RB or ADVP that serves to modify the meaning of the word. Unlike acomp and xcomp typed dependency relations, advmod relation is less likely to indicate the presence of a product feature specific opinion because of the absence of the verb, but more likely to indicate the presence of a general opinion because of the presence of the adjective or the adverb that modifies the adjective or the verb. When the governor component is an adjective and the dependent component is an adverb, advmod mostly indicates the presence of an opinion, and such combination can be found very frequently. Also, when both the governor component and the dependent component of the advmod typed dependency relation are adverbs, it does not represent any product feature functionality by itself. On the other hand, when the governor component is a verb, advmod relation quite often does not indicate the presence of an opinion, but the verb remains an indicator of a functionality of the product for which the reviewer expresses his opinion somewhere else in the sentence. It is needed to be mentioned that adjectival modifier (amod) typed dependency relation sometimes represents opinion and sometimes does not; thus could not be used as a definitive indicator to identify product feature specific opinion sentences. Table 4. shows examples taken from review lines in domain '*Electronics*' where advmod relation can possibly indicate the present of an opinion in a review sentence.

Table 4. Example of advmod as an opinion indicator

Dependency Relation	Component Example	Indication
advmod	well/JJ amazingly/RB	Possible Opinion
advmod	easiliy/RB very/RB	Possible Opinion
advmod	loads/VBD fast/RB	Possible Opinion
advmod	looks/VBZ especially/RB	Not Opinion
advmod	fits/VBZ perfectly/RB	Possible Opinion
advmod	recognized/VBN straight/RB	Not Opinion
advmod	satisfied/VBN very/RB	Possible Opinion
advmod	priced/VBN reasonably/RB	Possible Opinion

4.1.2Opinion Sentence Detection

When the above mentioned typed dependency relations are present in the review sentences, following algorithm has been used to determine whether a review sentence can be considered as a opinion sentence.

Figure 1. Algorithm to identify opinion sentences

```
for each sentence in review text
    set Opinion_Flag=False
    check acomp_presence
    if present
        if governor is any form of verb
            if dependent is any form of adjective
                set Opinion_Flag=True
    check xcomp_presence
    if present
        if governor is any form of adjective
            if dependent is any form of verb
                set Opinion_Flag=True
        else if governor is any form of verb
            if dependent is any form of adjective
                set Opinion_Flag=True
    check xcomp_presence
    if present
        if dependent is any form of adverb
            if governor in any form of verb
                set Opinion_Flag=True
            else if governor is any form of adverb
                set Opinion_Flag=True
            else if governor is any form of adjective
                set Opinion_Flag=True
```

4.2 Assigning Product Features

Each of the opinion sentences is assigned with a product feature with the help of the frequently associated words that appear with the selected typed dependency relations mentioned above.

4.2.1 Counting Frequent Words

As a product feature tag is assigned to each of the review sentences in the test data set, word counts are therefore done only within the product feature scopes. But instead of taking all the words of each sentence into consideration, only the words in component elements of the typed dependency relations are counted as they can be considered to carry the most indicative information to identify a product feature. Rest of the words in each sentence is ignored to avoid undesired words that do not relate to any specific product feature. Any word which is a function word is also ignored and is not involved in the counting process so that the common words that are present in any text can be avoided. While counting, lemmatization is used to consider only canonical form of the words so that the frequency of the words does not get distributed over different representations of same words.

If N is the total number of review lines present in the test data set and $p_1, p_2, ..., p_j \in P$ is the set of product features then word frequency count, WC_j for word w within p_j product feature scope can be denoted by the following equation:

$$WC_j = \sum_{i=1}^{N} w_{i,j}$$

where, $w_{i,j}$ is the frequency of the word w at review line i, associated with product feature, p_j. For different values of j, word frequency of the same word w will be different because associated product feature p_j will be different.

To include synonyms of the words in the counting process, Wordnet's synset for each word has been used. But because each of these words in synsets was not originally present in the review sentence, there is no surety that the synonym under consideration will be appropriate under the context. In addition to that, there can be more than one synsets in case of polysemous synonyms. Therefore, instead of counting each synonym for single occurrence, each of the synonyms is divided by the total number of synonyms found from all the synsets having the original word to represent a probability measure. That is, for k synsets having n_i synonyms in each, the probability of each synonym to be the appropriate synonym of the original word, w is considered by the following probability function:

$$P(w) = \frac{1}{\sum_{i=1}^{k} n_i}$$

This does not eradicate the noise in the word list introduced by polysemous synonyms, but minimizes the impact. This probability score is used as the frequency of the word synonyms.

4.2.2 Normalizing with tf.idf metric

To normalize the word frequencies, tf.idf metric has been used. If $WC_{i,j}$ is the frequency of word w_i in a product feature scope p_j, k is the number of total words in p_j, then term frequency, $tf_{i,j}$ can be denoted by,

$$tf_{i,j} = \frac{WC_{i,j}}{\sum_k WC_{k,j}}$$

if $|P|$ is the total number of product features assigned in the corpus, $|p : \{w_i \in p\}|$ is the number of product features with which the word w_i appears, then inverse document frequency idf_i can be calculated by the following,

$$idf_i = \log \frac{|P|}{s + |p : \{w_i \in p\}|}$$

The inverse document frequency calculation process suffers from a possibility of division by zero error. In the evaluation review data set, if there are new words that do not appear with any of the product features in the training data set, the denominator at the right side of the inverse document frequency calculation equation will have a zero value and *idf* cannot be calculated. To avoid this problem, a soothing parameter *s* has been used in the denominator. The value of *s* has been selected to be 0.001 which is a very small value that does not have any impact of its own in the calculation process. And finally, the tf.idf weight metric for word w_i can be calculated by multiplying term frequency $tf_{i,j}$ with inverse document frequency idf_i .

4.2.3Assigning Product Features

For each product feature, a product feature score is calculated using the following formula:

$$PFS = \sum f(acomp) + \sum f(xcomp) + \sum f(advm)$$

where *f(relation)* is a function that calculates the *tf.idf* weight score for each of the components of a typed dependency relation considering a specific product feature. *Tf.idf* scores for both the words at the governor and dependent position of the typed dependency relation is summed up for all the selected typed dependency relations that can be found in the sentence that is needed to be assigned a product feature. This score of each sentence is calculated for all the product feature classes.

PFS represents the contribution of a set of words in a sentence towards different product features. Because the *tf.idf* metric yields different scores for each of the words within different product feature scopes, *PFS* will have different values for each of the product features. When a product feature achieves a higher *PFS* value than the others, this means the words in the opinion sentence under consideration are more indicative of that product feature than of others. If *c* is the product feature class for which the product feature score, *PFS* is calculated, then each opinion sentence is assigned to a product feature class c^* where,

$$c^* = \arg\max_c PFS$$

From all the *PFS* scores calculated, a threshold value has been selected to be 1% of the highest *PFS* score. This is because some of the sentences that do not contain any opinion might have few words common with sentences that contain product feature specific opinions. But if not indicative enough, these words will yield a relatively low *PFS* score because they do not appear very frequently with the product feature specific opinion sentences. That is why, below this threshold value, *PFS* score is considered to be not strong enough to indicate a product feature and thus the corresponding sentence is considered as a not opinion bearing sentence.

5.Results

Manually annotated review data set of 50 reviews, kept for evaluation of the system, consisted of a total of 220 sentences having 113 opinion sentences and 107 sentences with no opinion. Table 5 shows sentence and word distribution of the selected product features in evaluation review set.

Table 5. Sentence and word distribution of test data

Product Feature	No. of Sentences	Average words per sentence (Without Function Words)	Average words per sentence (With Function Words)
General	57	5.63	7.77
Usability	16	11.44	13.69
Design	15	6.53	8.80
Portability	9	10.22	12.89
Performance	9	9.33	11.44
Speed	7	12.57	16.00

Based on the manually annotated test set of 50 test reviews in domain '*Electronics*' for product type '*hard disk*', the precision and recall scores for opinion detection are presented in Table 4.

Table 4. Evaluation score for opinion sentence detection

Precision	Recall	F-measure
0.7231	0.4159	0.5281

The evaluation scores for the assigned product features based on the manually annotated test set of 50 test reviews in domain '*Electronics*' for product type '*hard disk*' are presented in Table-5.

Table 5. Evaluation score of product feature assignment

Product Feature	Precision	Recall	F-measure
General	0.7778	0.1228	0.2121
Usability	0.9231	0.7500	0.8276
Design	0.6364	0.4667	0.5385
Performance	0.5833	0.7778	0.6667
Portability	0.7143	0.5556	0.6250
Speed	0.3077	0.5714	0.4000
No Opinion	0.5742	0.8318	0.6794

in the evaluation scores of product feature assignment, some of the product features achieved satisfactory result. This is because different verbs represent different functionalities of a product and assigned product feature to the opinion sentence. On the other hand, the opinion sentences that do not convey any product feature specific opinion; rather convey opinions of the reviewers in general categories are difficult to identify. As a result, the recall score is relatively low for general opinion sentences.

It has been observed that, quite often, a single sentence carries opinions about more than one product features. In this experiment, such sentences were tagged with only one product feature title. As a result, the words that are usually associated with the other product features but present in the same sentence contributed wrongly towards both. Appropriate segmentation methodology that can segment a single sentence in a way that only one product feature can be assigned to each sentence is needed to be applied in order to obtain a better result.

6.Conclusion

This paper discusses a process to detect opinion sentences and assigns a product feature to each opinion sentences. Typed dependency relations and frequent word associations have been utilized to achieve the desired goal. The obtained results leave room for improvement possibilities. Also, the process has been experimented within a very small scope. Future works will involve identifying appropriate segmentation methodology to aid the system, implementing the process in a number of varied domains and exploring left and right context of the dependencies for more supporting information towards product feature assignment.

7.References

[1] J. Wiebe. Learning Subjective Adjectives from Corpora. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, 2000.

[2] V. Hatzivassiloglou and J. Wiebe. Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In *Proceedings of the 18th conference on Computational linguistics*, Germany, 2000.

[3] P. D. Turney. Thumbs up or thumbs down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, 2002.

[4] N. Godbole, M. Srinivasaiah and S. Skiena. Large-scale Sentiment Analysis for News and Blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007.

[5] S.-M. Kim and E. Hovy. Automatic Detection of Opinion Bearing Words and Sentences. In *Companion Volume to the Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, 2005.

[6] E. Riloff and J. Wiebe. Learning Extraction Patterns for Subjective Expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2003.

[7] J. Wiebe, T. Wilson and M. Bell. Identifying Collocations for Recognizing Opinions. In *Proceedings of the ACL Workshop on Collocation: Computational Extraction, Analysis, and Exploitation*, Toulouse, France, 2001.

[8] Hong Yu and Vasileios Hatzivassiloglou. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2003.

[9] T. Wilson, J. Wiebe and R. Hwa. Just How Mad are You? Finding Strong and Weak Opinion Clauses. In *Proceedings of AAAI-04, 21st Conference of the American Association for Artificial Intelligence*, 2004.

[10] Z. Fei, X. Huang and L. Wu. Mining the Relation between Sentiment Expression and Target Using Dependency of Words. In *proceedings of 20th Pacific Asia Conference on Language, Information and Computation (PACLIC20)*, Wuhan, China, 2006.

[11] J. Yi, T. Nasukawa, R. Bunescu and W. Niblack. Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2003.

[12] J. Yi and W. Niblack. Sentiment Mining in WebFountain. In *Proceedings of the International Conference on Data Engineering (ICDE),* 2005.

[13] M. Hu and B. Liu. Mining Opinion Features in Customer Reviews. In *Proceedings of AAAI*, San Jose, USA, 2004.

[14] A.-M. Popescu, and O. Etzioni. Extracting Product Features and Opinions from Reviews". In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/ EMNLP)*, Vancouver, British Columbia, Canada, 2005.

[15] R. Ghani, K. Probst, Y. Liu, M. Krema and A. Fano. Text Mining for Product Attribute Extraction. SIGKDD Explorations Newsletter, 8(1). 2006.

[16] A. Qadir. Identifying Frequent Word Associations for Extracting Specific Product Features from Customer Reviews. In *Proceedings of International Symposium on Data and Sense Mining Machine Translation and Controlled Languages, and their Application to Emergencies and Safety Critical Domains*, Besancon, France, 2009.

[17] M.-C. de Marneffe and C. D. Manning. The Stanford typed dependencies representation. In *COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*, 2008.

[18] M.-C. de Marneffe and C. D. Manning. Stanford Typed Dependencies Manual. Technical report, 2008.

[19] L. Zhuang, F. Jing and X. Zhu. Movie Review Mining and Summarization. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, Arlington, Virginia, USA, 2006.

[20] T. H. King, R. Crouch, S. Riezler, M. Dalrymple and R Kaplan. The PARC 700 Dependency Bank. In *4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*, 2003.