A Comparative Study of Feature Extraction Algorithms in Customer Reviews

Liliana Ferreira
Institute of Electronics and Telematics
Engineering of Aveiro
University of Aveiro
Campus Universitário de Santiago,
3810-193 Aveiro, Portugal
lsferreira@ua.pt

Niklas Jakob and Iryna Gurevych
Ubiquitous Knowledge Processing Lab
Technische Universität Darmstadt
Hochschulstr. 10, 64289 Darmstadt, Germany
{njakob, gurevych}@tk.informatik.tu-darmstadt.de

Abstract

The paper systematically compares two feature extraction algorithms to mine product features commented on in customer reviews. The first approach [17] identifies candidate features by applying a set of POS patterns and pruning the candidate set based on the Log Likelihood Ratio test. The second approach [11] applies association rule mining for identifying frequent features and a heuristic based on the presence of sentiment terms for identifying infrequent features. We evaluate the performance of the algorithms on five product specific document collections regarding consumer electronic devices. We perform an analysis of errors and discuss advantages and limitations of the algorithms.

1 Introduction

The utilization of web communities as an information source has strongly increased over the past years. This trend was stimulated by the popularity of the integration of customer feedback in online shopping portals or service platforms. While the customers mostly desire to leave their feedback in a free and unstructured form, this kind of data is most difficult to process by software. Yet a lot of useful information can be found in customer reviews which are, on the one hand, beneficial for a potential customer by enhancing the purchase decision, and on the other hand valuable for a vendor since they contain free customer feedback.

One useful type of information available is the *opinion* people express about a given *subject*, being either a topic of interest or a feature of the topic. The interest in *opinion mining* on product reviews has increased over the last years [10, 14, 4, 13, 9]. The problem is typically decomposed into three main subtasks: (i) identifying topic specific features, such as product features, (ii) identifying opinions expressed about the features, and (iii) determining the sentiment orientation of the opinions.

This paper focuses on the first task, specifically extracting the product features in customer reviews. For this task, several approaches have been reported [14, 2, 12, 6]. Some of them rely on the calculation of the Point-wise Mutual Information between the given topic term and potential feature terms [14]. Other approaches require pre-built databases of feature terms [2, 6] or dynamically build such databases by extracting phrases which match predefined patterns [12]. To our knowledge, there exist two language independent approaches that do not rely on hand-crafted domain or world knowledge. Yi et al. [17] present the algorithm *Sentiment Analyzer* which identifies product features by extracting a set of base noun phrases as candidate feature terms and ranks them acccording to a relevance score. The evaluation is performed on two types of customer reviews: digital cameras and music review articles. For the digital camera domain, the authors report 100% precision when the Likelihood Ratio Test method is applied, but recall values are not reported. The experimental setup is not fully described, e.g. it is not completely clear which feature boundaries were used in the candidate feature terms extraction (see Section 4.3.1).

Hu and Liu [11] present a different approach to feature extraction. Their system uses association rule mining [1] to extract nouns as feature candidates occurring in product reviews. The data used to evaluate their system consists of reviews of consumer electronics from Amazon.com and C|net.com. An in-depth analysis of the data can be found in Section 2.

In this paper we focus on the approaches presented in [17] and [11], since they do not use any methods which require manually labelled training data and do not depend on any hand-crafted domain specific knowledge. We apply the two feature extraction algorithms described above to the data from [11]. We perform an additional annotation to study the performance of the feature extraction steps of each algorithm on the document and on an instance level. The remainder of this paper is organized as follows: Section 2 gives an overview of the data that has been used in the experiments. Section 3 introduces the algorithms employed in this study. Section 4 presents the evaluation and a systematic analysis of the errors made by the algorithms, and it discusses their possible improvements. Finally, the results are summarized in Section 5.

2 Data

We employed datasets of customer reviews for five products, collected from Amazon.com and C|net.com as described in [11]. These customer reviews focus on electronic products: two digital cameras, a DVD player, an MP3 player and a cell phone. Table 1 presents descriptive statistics about each dataset.

Table 1. Product review datasets

Dataset	Number of documents	Number of sentences
Digital camera 1 (DC1)	45	597
Digital camera 2 (DC2)	34	346
Cell phone (CP)	41	546
MP3 player (MP3)	95	1716
DVD player (DVD)	99	739

2.1 Annotation Scheme by Hu & Liu and Revised Annotation Scheme

Hu & Liu [11] define a product feature as a characteristic of the product which customers have expressed an opinion about, where an opinion is a statement which explicitly characterizes a feature in a positive or negative manner. Their annotation consists of the product feature(s) mentioned in the current sentence, where a feature is only annotated as such if an opinion is stated about it. For instance in the sentence:

(1) at the same time, i wanted my wife to not be intimidated by knobs and buttons.

no features are annotated, although the product features `knobs` and `buttons` are mentioned. Since we focus on the feature extraction step, we consider it necessary to annotate features in neutral sentences which contain product features, such as sentence 1.

In the revised annotation scheme, each entity to be annotated as a feature must satisfy one of the following criteria:

- *Part-of* relationship with the product the document is about; for example in the domain of digital cameras `battery` would be annotated as a feature of a camera.

- *Attribute-of* relationship with the product; for example `weight` and `design` would be considered as attributes of a camera.

- *Attribute-of* relationship with a known feature of the product of the document; for example `battery life` would be considered an *attribute of a feature* of the camera, specifically an attribute of the `battery`.

For example, in the sentence:

(2) the lens is visible in the viewfinder when the lens is set to the wide angle , but since i use the lcd most of the time , this is not really much of a bother to me.

the features `lens, viewfinder` and `lcd` are annotated in our annotation scheme, but not by Hu & Liu [11].

Table 2 presents comparative statistics based on the data annotated according to the original and revised annotation schemes. The second column gives the total number of distinct features annotated in each set of documents of the review data. Column 4 shows the number of distinct features found in the revised anotation. Columns 3 and 5 contain the number of annotated features where every instance of a product feature is counted.

Table 2. Number of features in original and revised annotation

	Original Annotation		Revised Annotation	
Dataset	Distinct	Total	Distinct	Total
DC1	99	257	161	594
DC2	74	185	120	340
CP	109	310	140	471
MP3	180	736	231	1031
DVD	110	347	166	519

We observe that the revised annotation contains far more features than the original annotation. This was to be expected since we annotated features irrespectively of an opinion being expressed about them or not.

3 Feature Extraction Algorithms

3.1 Likelihood Ratio Test Approach

The system described by [17] extracts features and their respective sentiment orientation from given documents. Determining the feature terms includes the following steps: (i)

Selecting candidate features terms, (ii) calculating a relevance score for each feature candidate term,[1] and (iii) identifying feature terms from the candidate feature terms based on the relevance scores.

1. Candidate Feature Term Selection: The heuristics used to select the candidate feature terms identify base noun phrases according to the following patterns:
Base Noun Phrase (BNP). This pattern restricts the candidate feature terms to one of the following patterns: *NN, NN NN, JJ NN, NN NN NN, JJ NN NN, JJ JJ NN*, where *NN* and *JJ* are nouns and adjectives.
Definite Base Noun Phrase (dBNP). This pattern restricts candidate feature terms to definite base noun phrases, which are noun phrases (*BNP*) preceded by the definite article *the*.
Beginning Definite Base Noun Phrase (bBNP). *bBNP*s are *dBNP*s at the beginning of a sentence followed by a verb phrase.

2. Relevance Scoring: The feature weighting algorithm applied in [17] is based on the Likelihood Ratio test [5].
Likelihood Ratio Test: Let D_+ be a collection of documents dealing with a topic T and D_- a collection of documents not about T. A *BNP* is a candidate feature term occuring in D_+. The likelihood ratio $-2\log\lambda$ is then defined as:

$$-2\log\lambda = \begin{cases} -2 * lr & \text{if } r_2 < r_1 \\ 0 & \text{if } r_2 \geq r_1 \end{cases} \tag{1}$$

$$\begin{aligned}
r_1 &= \frac{C_{11}}{C_{11}+C_{12}}, r_2 = \frac{C_{21}}{C_{21}+C_{22}} \\
r &= \frac{C_{11}+C_{21}}{C_{11}+C_{12}+C_{21}+C_{22}} \\
lr &= (C_{11}+C_{21})\log(r) + (C_{12}+C_{22})\log(1-r) - C_{11}\log(r_1) \\
&\quad - C_{12}\log(1-r_1) - C_{21}\log(r_2) - C_{22}\log(1-r_2)
\end{aligned}$$

C_{11} to C_{22} are defined in Table 3.

Table 3. Counting a BNP

	D_+	D_-
BNP	C_{11}	C_{12}
$\overline{BNP}$	C_{21}	C_{22}

The higher the value of $-2\log\lambda$, the higher the likelihood that the *BNP* is relevant to the topic T.

3. Feature Identification: For each *BNP*, we compute the likelihood ratio score $-2\log\lambda$, as defined in Equation 1. Then we sort the *BNP*s in decreasing order of their likelihood score. Feature terms are *BNP*s whose likelihood ratios satisfy a predefined confidence level. Alternatively, the top n *BNP*s can be selected [17].

3.2 Association Mining Approach

The goal of the work by Hu and Liu [11] is to automatically create summaries of customer reviews. Hu & Liu assume that the product features appear as nouns and that the opinions about these features are expressed by adjectives. A distinction is made between so called *frequent features (ff)* and *infrequent features (iff)*. *Frequent features* appear in several documents, while *infrequent features* are commented on less often.

1. Identifying Frequent Features and Feature Sets: Association mining [1] is employed in order to extract the *frequent features*. The association mining algorithm calculates the probability that certain features or feature sets occur in the review document collection for a certain product. Candidate terms for both kinds of features are nouns only. The nouns occurring in a sentence are used to create a so called *transaction set*. The transaction sets from all reviews of a certain product are input to the association mining algorithm. A certain feature or feature set is considered frequent if its *minimum support* is larger than an empirically defined threshold of 1%. Minimum support is defined as the minimum percentage of transaction sets that contain all of the features listed in that association rule.
Since association mining does not consider the position of the terms in sentences, two pruning steps are applied: The first pruning step is called *compactness pruning*. It removes *frequent feature sets (ffs)* in which the individual terms do not occur within a distance of three or less words in two or more sentences of the document collection. The second pruning step called *redundancy pruning* removes *ff* or *ffs* which are complete subsets of other *ffs*, if the subset does not occur by itself in three or more sentences.

2. Identifying Opinions: Identifying opinions about the product features follows a lexicon based approach. Based on previous work on the correlation of subjectivity and the presence of adjectives in sentences [3, 16], opinion words are assumed to be adjectives. The lexicon of opinion words is created by crawling WordNet [7] starting from seed adjectives, see Table 4. By crawling synonyms and antonyms of the seed adjectives in WordNet, we create a list with 99 positively and 111 negatively oriented adjectives.

[1] Yi et al. compute the relevance scores using the *Likelihood Ratio Test* [5] and the *Mixture Model* method. Since the *Likelihood Ratio Test* consistently outperformed the *Mixture Model* method, we focus on the former one in the present study.

Table 4. Seed terms for opinion lexicon

positive	*negative*
happy, great, fantastic, nice, cool, awesome, beautiful, perfect, excellent, intuitive, super, superb	bad, dull, horrible, poor, terrible, weak, ugly, difficult, unsatisfactory, disappointing

3. Identifying Infrequent Features: *Infrequent features (iff)* are extracted from the sentences which do not contain any *ff*s, but contain an opinion word. In this case, the noun(s) with the smallest distance (in words) to the opinion word are extracted. The *iff* identification step is reported to increase the average recall by 0.13 to 0.80 with a precision decrease of 0.07 to 0.72 in [11].

Table 5. Comparison of approaches to product feature identification in customer reviews

	Likelihood Ratio Test Approach	Association Mining Approach
Candidate feature extraction	Patterns of POS sequences. Probabilities in specific and general domain corpora	Nouns and noun sets depending on their minimum support + *iff*s
Depends on opinion identification	No	Partly
Uses empirically defined threshold	Yes, for Likelihood Test	Yes, for *minimum support*
Considers position of feature in a sentence	Yes	Partly with compactness pruning
Can extract multi-word features	Yes	Yes
Requires general vocabulary corpus	Yes	No

3.3 Comparison of the Approaches

Table 5 presents a comparison of the two approaches, summarizing the methods used by each of them. We observe that the Association Mining approach is less restrictive in the selection and extraction of candidate features. As outlined in Section 3.1, the *BNP* patterns restrict candidate terms for multi-word features to consecutively occurring nouns, whereas the Association Mining approach can combine nouns occurring anywhere in a sentence to a multi-word feature. This characteristic of the association mining creates more flexibility compared to the Likelihood Ratio Test approach concerning multi-word feature extraction, but at the same time introduces a new source of potential errors. Therefore the employment of the compactness pruning step is necessary. Both approaches rely on a threshold which affects the feature selection, for which it is not possible to calculate an ideal value in advance.

4 Evaluation

4.1 Experimental Setting

Setting for the Likelihood Ratio Test Approach: As a collection of topical documents (D_+) we employ the product review datasets described in Table 1. As non-topical documents (D_-), approximately 600 documents were randomly selected from the UKWaC British English web corpus [8]. We ran the feature extraction algorithm with three different methods extracting either: *BNP*s (BNP-L), *dBNP*s (dBNP-L) or *bBNP*s (bBNP-L). For POS tagging, *TreeTagger* [15] is employed, which was not retrained for our datasets, but instead used with the provided default english parameter file.

In order to make the results of the three methods comparable, we extracted the same number of features with each of them. We therefore employ all three methods and set the likelihood threshold to 0 for each of them. The *bBNP-L* method will always extract the fewest number of results, since its candidate *BNP*s are a subset of the ones extracted by the other two methods. We therefore only use the top n features extracted by the *BNP-L* and *dBNP-L* method for the evaluation, where n is the number of *BNP*s extracted by the *bBNP-L* method. The results of this evaluation are shown in columns 3 and 4 of Table 7.

Setting for the Association Mining Approach: Since association mining disregards the original ordering of the terms in sentences, we cannot reconstruct whether the extracted *ffs* [picture, quality] occurred as "quality picture" or "picture quality" in the dataset. For the evaluation, we therefore match every permutation of an extracted *ffs* against a multi-word feature in the annotation. If one term order results in a match, we count that as a correct result, otherwise it is considered a false result. If the returned feature is just a subset or subsequence of the annotated feature we consider that a false result too.

4.2 Evaluation Methods and Results

We evaluate the feature extraction algorithms described in Sections 3.1 and 3.2 with two different methods. The first one (eval-1) examines how well the algorithms perform on the task of extracting features which were commented on in the entire collection of reviews. This evaluation strategy corresponds to the task of creating a summary of features for the review collection as a whole. The second evaluation (eval-2) studies the performance of the algorithms on an instance level, where each feature extraction is counted individually. The original annotation scheme by Hu & Liu does not cover product features with neutral orientation. As we are interested in identifying product features irrespectively of the opinion expressed, we base the evaluation on the revised annotation scheme only.

4.2.1 Document Collection Level Evaluation

In the document collection level evaluation the algorithms extract a list of distinct features from the entire document collection of product reviews.Columns 5 and 6 of Table 6 present the results obtained with the *dBNP-L* method, which outperforms the *BNP-L* and *bBNP-L* methods. This is different from what has been reported in [17], where the best performance was obtained with the *bBNP-L* method. Precision values are substantially higher than recall values, with an average of 80% precision and 10% recall. For the collection of digital camera reviews, which yielded best results in [17] (precision of 100%), we achieved an average precision of 80% and 16% recall. Columns 3 and 4 of Table 6 give the results yielded by the Association Mining approach. We observe that both average recall and precision values are fairly low. Recall is rather low since the association mining always fails to correctly extract certain features due to the threshold employed or due to the pruning steps. Precision is low due to the fact that the association mining algorithm is not capable of distinguishing between the correct nouns related to the current product and the nouns belonging to the general vocabulary.

4.2.2 Instance Level Evaluation

Since an evaluation on the product specific document collection is targeted at extracting a summary of product features, which does not take into account the frequency of individual features being discussed, we also conduct an instance level evaluation. For each sentence, we compare the annotated feature(s) to the feature(s) extracted by the algorithms. Table 7, columns 3 and 4 present the results of the Likelihood Ratio Test algorithm based on different configurations described in Section 3.1. The best recall values are always achieved with the *BNP-L* method, while the precision is higher with the *dBNP-L* method, except for the MP3 documents where the highest precision is achieved with the *bBNP-L* method.
Table 8, columns 3 to 8 present the results obtained with the Association Mining approach. Columns 9 and 10 display the results obtained with the Likelihood Test approach (*dBNP-L*). We observe that in the instance level evaluation, the recall values of the association mining algorithm are considerably higher compared to the document collection level evaluation, while the precision is moderately lower. For the Likelihood Ratio Test algorithm, the tendency to higher recall and lower precision is also observed with recall improving by 9% and precision decreasing by 12%.

Table 7. Instance level results for different Likelihood Ratio Test methods. Results with and without subsequence similarity (SsS)

		Without SsS		With SsS	
		Recall	Precision	Recall	Precision
DC1	*BNP-L*	0.456	0.618	0.495	0.671
	dBNP-L	0.256	0.798	0.271	0.846
	bBNP-L	0.039	0.719	0.044	0.812
DC2	*BNP-L*	0.323	0.590	0.347	0.634
	dBNP-L	0.156	0.776	0.159	0.791
	bBNP-L	0.024	0.533	0.039	0.867
CP	*BNP-L*	0.406	0.583	0.459	0.659
	dBNP-L	0.197	0.742	0.212	0.798
	bBNP-L	0.043	0.667	0.049	0.767
MP3	*BNP-L*	0.364	0.302	0.408	0.339
	dBNP-L	0.254	0.473	0.263	0.490
	bBNP-L	0.061	0.596	0.072	0.702
DVD	*BNP-L*	0.165	0.344	0.243	0.506
	dBNP-L	0.107	0.647	0.132	0.800
	bBNP-L	0.016	0.571	0.021	0.786

4.2.3 Comparison of the Evaluation Strategies

In the document collection level evaluation it does not matter from which sentence and document an algorithm extracts a certain feature, it is important that the feature is found at least once. The total number of targeted features is considerably lower than in the instance level evaluation and therefore the decrease in recall is in turn higher if an algorithm fails to extract a certain feature. Comparing the results of the two evaluation methods however indicates that these problematic features seem to occur rather seldomly throughout the entire document collection, since the average recall of the instance level evaluation is higher than the recall of the document collection level evaluation.

4.3 Error Analysis

In this section, we analyze the sources of errors identified in the output of the algorithms. Table 10 gives a classification of the errors of both algorithms, for the DC1 document collection. Table 9 lists the top 20 features terms extracted from the DC1 customer reviews by the two algorithms. Some of these terms, like `nikon coolpix, week, work` are wrongly classified by the Likelihood Ratio Test approach as product features. The association mining algorithm falsely extracts `week, box, way, work` as product features. A discussion of error sources associated with the Likelihood Ratio Test approach is done in Section 4.3.1, in Section 4.3.2 we analyze the errors of the Association Mining approach and in Section 4.3.3 we compare the two approaches.

Table 6. Feature extraction results on document collection level

Dataset	All distinct features	Asso. Mining approach			Likelihood Test approach		
		ff + *iff* extraction			*dBNP-L*		
		Recall	Precision	F-measure	Recall	Precision	F-measure
DC1	161	0.363	0.318	0.339	0.118	0.864	0.208
DC2	120	0.337	0.225	0.270	0.100	0.923	0.180
CP	140	0.339	0.500	0.404	0.114	0.889	0.202
MP3	231	0.216	0.433	0.288	0.138	0.615	0.225
DVD	166	0.254	0.358	0.297	0.048	0.727	0.090
Average	**164**	**0.302**	**0.367**	**0.320**	**0.104**	**0.804**	**0.181**

Table 8. Feature extraction results on instance level

Dataset	All Features	Association Mining approach						Likelihood Test approach		
		ff extraction			*ff* + *iff* extraction			*dBNP-L*		
		Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
DC1	594	0.604	0.307	0.407	0.614	0.308	0.410	0.252	0.798	0.383
DC2	340	0.652	0.295	0.406	0.661	0.296	0.409	0.156	0.776	0.260
CP	471	0.530	0.312	0.393	0.535	0.310	0.393	0.197	0.742	0.311
MP3	1031	0.523	0.197	0.286	0.531	0.195	0.285	0.254	0.473	0.331
DVD	519	0.483	0.223	0.305	0.491	0.223	0.307	0.107	0.647	0.184
Average	**591**	**0.558**	**0.266**	**0.359**	**0.566**	**0.266**	**0.361**	**0.193**	**0.687**	**0.294**

Table 9. Top 20 features according to their rank

	DC1
bBNP-L	camera, nikon, digital camera, picture, canon, battery, g3, lens, flash, lcd, photo, battery life, viewfinder, picture quality, feature, shutter, nikon coolpix, quality, shot, optical zoom
Asso. mining	canon, g3, canon g3, powershot g3, purchase, camera, camera g3, week, picture, picture camera, box, way, work, g2, quality, picture quality, setting, flash, card, feature

4.3.1 Analysis of the Likelihood Ratio Test Approach

The main problem of this approach is the low recall. There exist several reasons for that. The first one is related to the threshold set by the *bBNP-L* method, which is in turn used in the *bBNP-L* method to limit the number of extracted features (see Section 3.1). *bBNP*s are base noun phrases preceded by a definite article in the beginning of a sentence, and followed by a verb phrase. However, the product features are only seldomly preceded by the determiner *the*, especially in the beginning of a sentence. For instance, in the DC1 dataset, only 32 distinct *bBNP*s with likelihood value larger than 0 were extracted from 161 distinct features. Too many extracted *dBNP*s and *BNP*s are not considered. Another problem is related to the extraction of feature terms which occur both in the general vocabulary and a domain specific vocabulary like e.g.:

(3) `box, filter, option, video, dial, flexibility, automation, speed` (DC1)

The Likelihood Ratio Test algorithm computes a probability score for each candidate feature term using the information about the number of occurrences in the topical (D_+) and in the non-topical documents (D_-). These terms display a relatively high number of occurrences in both types of documents. As the algorithm only extracts terms with a high probability of being product features, it will not extract features which are also common vocabulary terms.

Algorithm Modifications: The algorithm described in [17] does not cover the identification of feature boundaries for *BNP*s and *dBNP*s. Candidate feature terms are restricted to base noun phrases matching one of the patterns listed in Section 3.1. However, it is not defined which pattern should be used if there are multiple matches. For instance, in the expression `battery life` three candidate features can be considered: `battery life`, `battery` or `life`, resulting in low precision. Therefore we modify the algorithm in order to extract only the terms matching the longest *BNP* pattern.

The second modification is applied because many of the candidate *BNP*s are a combination of adjectives and nouns. For instance, in the expression `great photos`, which matches the BNP pattern *JJ NN*, the correct feature term is only the noun `photos`. To address this problem, we modi-

Table 10. Overview of different error classes in DC1

Algorithm	All Features	Multi-word features					Single-word features			
		False negatives			False positives		False negatives		False positives	
		None of the terms extracted	Some of the terms extracted	All extracted but not combined	On-topic features	Off-topic features	Not extracted	Extracted falsely as multi-word	On-topic features	Off-topic features
Association Mining	594	14	62	25	23	0	45	39	260	259
Likelihood Test	594	138	1	0	1	0	295	3	29	0

fied the algorithm in order to consider only the subsequence of the extracted feature which consists of nouns. We refer to these two modifications as *Subsequence Similarity* (SsS). An evaluation of these modifications is shown in columns 5 and 6 of Table 7. We observe an average increase of recall by 2% and an increase of precision by 10%.

4.3.2 Analysis of the Association Mining Approach

The precision of the Association Mining approach is fairly low, since it returns any noun as a feature if it often occurs in the documents. For example the term `week` is extracted as a frequent feature. There is no distinction between dataset specific terms and common vocabulary terms. Setting the *minimum-support* threshold higher will not solve this problem, as it would lead to decreased recall. Note that in our evaluation (Table 8) the infrequent features hardly affect the algorithm's results since they are only very seldomly extracted at all. For example in the DC1 dataset, of the 597 sentences only 12 contain an opinion word but no frequent feature. Of those 12 cases the infrequent feature identification leads to 7 correct and 5 false features being extracted.
In some cases (see Column 9 of Table 10), the association mining falsely attributes nouns occurring in a sentence to a single feature set. For example in

(4) recent price drops have made the g3 the best bargain in digital cameras currently available.

`[g3, camera]` is extracted as a feature set, since the two terms occur together as one entity in multiple other sentences. The compactness pruning will therefore not remove this feature set. Sentences as 4 will hence result in an error during extraction. The large amount of false positives in the single-word feature extraction (see Table 10 Columns 6 & 7) is due to the fact that many sentences in the DC1 dataset consist of comparisons of the DC1 camera to other camera models. The features of these other camera models are also mentioned in the reviews and therefore falsely extracted by the association mining, since the algorithm is not capable of distinguishing between references to features of the DC1 camera and any other camera model.

4.3.3 Comparison of the Approaches

As outlined in Table 10 the two approaches have their strengths and weaknesses in different tasks. If the Likelihood Ratio Test approach fails to extract a multi-word feature, the tendency is that none of the feature terms are being extracted, while this is not the case in the association mining approch. This is due to the fact that the association mining algorithm will return any feature combination occurring in a given sentence, while the Likelihood Ratio Test approach requires that a multi-word feature occurs in the same ordering in several sentences, in order to achieve a high likelihood ratio and therefore be extracted. The threshold of the Likelihood Ratio Test approach in combination with the *Subsequence similarity* calculation will therefore prevent that a subset of a multi-word feature is extracted, instead the feature will not be extracted at all. At the same time the association mining extracts several false multi-word features, none of them belonging to the general vocabulary.
We observe similar results in the analysis of the single-word errors. The Likelihood Ratio Test approach fails to extract many of the features, which is again due to the threshold, while the Association Mining approach extracts less false features, but has the problem of wrongly extracting actual single-word features as a multi-word expression as analyzed in Section 4.3.2. The inability of the Association Mining approach to recognize whether a certain candidate feature is an attribute of the current topic, as defined in Section 2.1, is observable in Columns 10 & 11 of Table 10. The Association Mining approach extracts a large number of false features compared to the Likelihood Ratio Test approach. The low number of falsely extracted on-topic features of the Likelihood Ratio Test approach could be attributed to the *dBNP* method. Apparently, if a candidate *BNP* is preceded by a definite article, an on-topic feature follows. However, the low number of false positives during the feature extraction reflects the tradeoff between recall and precision of this approach.

5 Conclusions

In this paper we provide a comprehensive analysis of two state-of-the-art algorithms for extracting features from product reviews based on the Likelihood Ratio Test and on association mining. The Likelihood Ratio Test fails to extract features also belonging to common vocabulary and it makes the extraction dependent on the feature position in the sentence, leading to low recall. The *dBNP* and *bBNP* based methods yield low recall due to the fact that the product features do not occur with the article *the* in front of them very often.

The Association Mining approach returns all frequent nouns, which decreases precision. Our results suggest that the choice of algorithm to use depends on the targeted dataset. If it consists of mainly on-topic content, the results of Table 10 indicate that the Association Mining algorithm is better suited for this task, due to its high recall. If the dataset consists of a mixture of on- and off-topic content, our results suggest that the Likelihood Ratio Test based algorithm would perform better, due to its ability to distinguish and filter out the off-topic features. For future work, we plan to extend the Likelihood Ratio Test methods, especially the *dBNP* based approach, by other determiners such as *a* or *this*, which should increase the recall of this method. Another possibility which we will investigate regards the *BNP* patterns. The current Likelihood Ratio Test approach is not capable of dealing with discontinuous feature phrases for example in:

(5) the quality of the pictures is great.

the feature would be `picture quality`. This problem could be addressed by introducing wildcards in the *BNP* patterns. We will also investigate whether there are any methods in order to calculate an optimal threshold for the candidate feature extraction, in order to increase the recall of the Likelihood Ratio Test based algorithm. We plan to investigate whether a deeper linguistic analysis, e.g. with a dependency parser, can improve the feature extraction.

Acknowledgements The project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ07012. The authors take the responsibility for the contents.

The information in this document is proprietary to the following Theseus Texo consortium members: Technische Universität Darmstadt. The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. Copyright 2008 by Technische Universität Darmstadt.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, 1215:487–499, 1994.

[2] K. Bloom, N. Garg, and S. Argamon. Extracting appraisal expressions. In *HLT-NAACL 2007*, pages 308–315, 2007.

[3] R. Bruce and J. Wiebe. Recognizing subjectivity: a case study in manual tagging. *Natural Language Engineering*, 5(02):187–205, 1999.

[4] K. Dave, S. Lawrence, and D. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*, pages 519–528, New York, NY, USA, 2003. ACM.

[5] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[6] O. Feiguina and G. Lapalme. Query-based summarization of customer reviews. In *Canadian Conference on AI*, pages 452–463, 2007.

[7] C. Fellbaum. *Wordnet: An Electronic Lexical Database*. MIT Press, 1998.

[8] A. Ferraresi. Building a very large corpus of english obtained by web crawling: ukwac. Master's thesis, University of Bologna, Italy, 2007.

[9] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining customer opinions from free text. In *Proceedings of the 6th International Symposium on Intelligent Data Analysis (IDA-2006)*. Springer-Verlag, 2005.

[10] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo. Deriving marketing intelligence from online discussion. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 419–428, New York, USA, 2005. ACM.

[11] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of 9th National Conference on Artificial Intelligence*, 2004.

[12] N. Kobayashi, K. Inui, K. Tateishi, and T. Fukushima. Collecting evaluative expressions for opinion extraction. In *Proceedings of IJCNLP 2004*, pages 596–605, 2004.

[13] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima. Mining product reputations on the Web. In *Proceedings of KDD-02, 8th ACM International Conference on Knowledge Discovery and Data Mining*, pages 341–349, Edmonton, CA, 2002. ACM Press.

[14] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of HLT-EMNLP-05, the Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing*, pages 339–346, Vancouver, CA, 2005.

[15] H. Schmid. Treetagger a language independent part-of-speech tagger. *Institut fur Maschinelle Sprachverarbeitung, Universitat Stuttgart*, 1995.

[16] J. Wiebe, R. Bruce, and T. O'Hara. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 246–253. Association for Computational Linguistics Morristown, NJ, USA, 1999.

[17] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceeding of ICDM-03, the 3ird IEEE International Conference on Data Mining*, pages 427–434, Melbourne, US, 2003. IEEE Computer Society.