

Feature Extraction from Product Reviews using Feature Similarity and Polarity

Alejandra Lopez
Fernandez
UCD School of Computer
Science and Informatics
Dublin, Ireland
alejandra.lopez-
fernandez@ucd.ie

Tony Veale
UCD School of Computer
Science and Informatics
Dublin, Ireland
tony.veale@ucd.ie

Prasenjit Majumder
UCD School of Computer
Science and Informatics
Dublin, Ireland
prasenjtit.majumder@ucd.ie

Technical Report
UCD-CSI-2009-09
October 2009

ABSTRACT

Research on developing techniques to access user generated content, and specifically user reviews on different products, came in the focus of the information research community in recent past. In particular, this paper addresses the problem of extracting the features from user comments of a particular product, taking advantage of a corpus with a semi-structured format: pros, cons and summary.

In this paper we propose a technique to extract a set of features based on user generated pros and cons for a particular product. Then using this set we test a feature similarity function to obtain new features from reviews (both from the pros/cons and from the free text summary) of the same and other products. Our experimental results have shown interesting conclusions.

General Terms

Feature extraction, opinion mining, feature similarity function.

1. INTRODUCTION

Information can be broadly classified into two categories; factoid and opinionated. Factoid information can be ranked whereas the later cannot be. Opinions or reviews can be stated as a interpretation of fact by an individual. Thus every opinion comes with its own subjectivity.

With the explosive growth of the Web, from the last decade, availability of user generated content increased immensely. Personal blogs, e-forums, the Yahoo! and Google groups, Amazon.com and CNet are some examples of this fact. The

last two sources explicitly contain user reviews on different products. These information sources are real treasure for companies wanting to understand the impact of their product on the end users. These opinions may even influence the manufacturer to re-design a product to make it more popular.

Research in developing techniques to access such information sources automatically and meaningfully came in the focus of the information research community in recent past. This area lies on the cross roads of IR and NLP research. Opinion mining research mainly address sentiment classification and information summarization in different granularity e.g document or sentence level. Classification of a subjective document based on its appreciation to a concept, product, or policy is a new type of text classification problem.

The classification can be assigned at different levels. e.g., given a movie review, the system determines whether the review expresses positive or negative opinion of the reviewer towards the movie. This is treated as a document classification problem and it could be a two-class [12] or multi-class [10] classification task. In the next level the sentence are analysed to identify the features of the object and whether the reviewers have liked or disliked a particular feature of the object. An object can be a product, a movie, an organization, a government policy, etc.

Direct comparison of one object with another of similar kind from the user opinions is gaining popularity [9] e.g., "which will be suitable in my case, Intel or AMD?". Thus qualitative comparison between two products in a given feature space or even sometimes ranking an object as the best or worst among a set of similar products e.g., "Overall performance of the Toshiba lap-tops is the best", is gaining popularity among the buyers.

Several feature extraction methods are available in the literature [1, 2, 3, 5, 6, 7, 9, 13, 14]. In this paper we propose a technique to extract a set of features from pros and cons for a particular product. Then using this set we test a feature similarity function to obtain new features from reviews of

the same and other products (both from pros/cons and free text) . We initially tried SentiWordnet [4], to classify the polarities, but by manual inspection it was found that SentiWordnet missed a considerable number of good features. This motivates us to use word association from the Google N-grams to populate more features. In a subsequent experiment we tried to extract features from one type of consumer electronic gadget reviews using features of another type of similar gadget. Thus an automatic cross-product feature extraction method is tried and evaluated in this process.

The rest of the paper is organised as follows. The next section will describe the new corpus, while Section 3 will focus on the feature extraction process we adopted here. The cross-product feature extraction and evaluation is given in the last two sections respectively.

2. CORPUS

We want to pursue two long term objectives in opinion mining research. First, to check the consistency between the written reviews and the user marking process, e.g. the review text may praise a lot but while giving scores the users might be more conservative. To understand this behavior we need a corpus with product reviews along with their scores. Second, to generate text summaries from multiple opinions about a particular product and to evaluate that summary with the editor’s written one. This motivates the generation of a big corpus from online product reviews available from the web in *Format1*[8]. Later, we will supplement the corpus with the reviews of the same products in *Format2*. *Format1* has some categories like pros, cons and summaries. Since we do not intend to find the polarities in this paper, but to try a new feature similarity function, we constructed this corpus.

In [11], several publicly available annotated datasets which can be a source of opinion, sentiment, and subjectivity labels are described. The most relevant to our task is perhaps, *Customer review datasets*¹, where the specific product features are labeled with an associated polarity (positive or negative and its strength). Unfortunately this corpus is rather small, 9 products, and it has been manually labeled.

We have chosen CNet as our source of reviews, which includes user reviews for electronic products. These reviews are in *Format1*. In general, the pros and cons are given in a concise format, in many cases in the form of a list of phrases separated by comma, or other delimiters. These phrases are generally in the format: qualifier + product feature. The polarity of those is implicit, it can be inferred that they are positive opinions if they are pros, and negative opinions if they are cons.

2.1 Corpus construction

The corpus construction can be divided into three subtasks: first, the selection of suitable sites to crawl; second, obtaining the appropriate content from the Web; and third giving a convenient format to the collection. These tasks are described below in further detail. We selected *CNet* as a source to get the user reviews. We wanted to take advantage of

¹<http://www.cs.uic.edu/?liub/FBS/CustomerReviewData.zip>

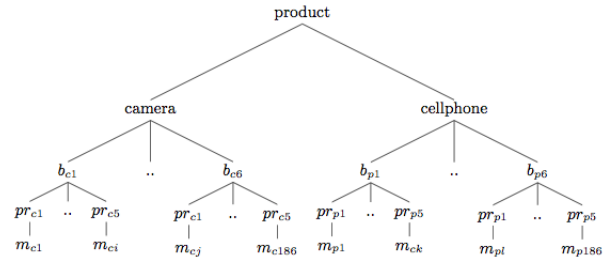


Figure 1: Corpus organization. b =brand, pr =price range, m =model, c =camera and p =cellphone.

the structure of its reviews in *Format1*[8]. We selected digital cameras and cellphones as initial products. For each product, 186 models were selected, within these 6 brands and 6 price categories (US\$100-US\$1200). Table 1 shows a summary of the distribution of the models in the corpus according to brands.

Camera		Cellphone	
Brand	#Model	Brand	#Model
Sony	29	Samsung	46
Panasonic	15	Nokia	21
Nikon	33	Sony Ericsson	7
Kodak	15	Motorola	29
Canon	54	BlackBerry	20
Olympus	11	LG	29
Others	29	Others	34

Table 1: Distribution of models by brands.

The models with at least 10 user reviews, and with the editor evaluation, were selected. A customized site crawler was developed for this purpose. Corpus cleaning and re-formatting follows the crawling. Figure 1 shows the structure of the data that was collected. One single document in xml format for each model of camera m_{ci} , and cellphone m_{pj} contains the most relevant information of the product: name, specifications, editor review, rate, and user reviews, etc. A typical document of the corpus is shown in Figure 2.

2.2 Analysis of corpus

In this section, we discuss some statistics of the CNet collection, as presented in Table 2. There are almost two times more reviews for cellphones than cameras, so we selected the cellphone reviews as the test corpus. Both corpora keep consistency in size, having in average almost the same number of words. All reviews have title, and very few do not have pros or cons. Around 20% of the reviews do not have a summary.

```

<product name="Olympus Stylus 800">
  <productSummary></productSummary>
  <specifications></specifications>
  <whereToBuy></whereToBuy>
  <similarProducts></similarProducts>
  <editorReview></editorReview>
  <userReviews>
    <review id="1496653">
      <usefulness>2 out of 3</usefulness>
      <userRate>1.0 stars</userRate>
      <title>Bad bad bad</title>
      <author xhref="/community/X">
        X
      </author>
      <date> October 25, 2005 </date>
      <pros>
        Fast, water resistant, compact
      </pros>
      <cons>
        Image quality
      </cons>
      <summary>
        I have no idea what camera the "9" rating users have but either
        it isn't this camera or they aren't using it for anything more
        than low resolution pictures in a slideshow on their computers.
        The camera over sharpens photos to a drastic extent. Looking
        at the images at 100% shows serious pixelation and sharp lines.
        It seems that the camera has way too much in-camera sharpening
        and noise reduction. On the subject of noise reduction, any
        mode other than automatic at 64 - 200 ISO at SHQ fails miserably.
        The "scene" modes are terrible, as they knock down and change
        ISO dramatically. And if you want low light pictures, forget
        about it. All you'll see is grain.
      </summary>
    </review>
  </userReviews>
</product>

```

Figure 2: Sample markup of a review from the collection.

Camera	Total #	Average # words
#Reviews	6480	144.3
#Titles	6480	5.6
#Pros	6345	14
#Cons	6307	12.5
#Summaries	4928	148.1
Cellphone	Total #	Average # words
#Reviews	13024	166.5
#Titles	13024	5.6
#Pros	12859	13.7
#Cons	12841	16.2
#Summaries	10994	155.6

Table 2: The main statistics of the collection.

3. FEATURE EXTRACTION

As we have mentioned, in this research we are studying the problem of extracting the features that customers have expressed about a particular product, taking advantage of a corpus with a semi-structured format, or *Format1* [8]. The feature extraction process is performed in three stages:

1. Features are first extracted from the pros and cons fields of the camera reviews.
2. These preliminary features are used to extract new candidate features from the summaries of the camera reviews, selecting those which are similar to the preliminary ones. This task can be seen like extraction from free text.
3. The final task is done by using the same similarity function and the features of a particular product (digital cameras), to extract features in other products (cellphones).

We used the digital camera reviews as the training corpus, and the cellphone reviews are used to test this last stage. In the following sections, each of the mentioned aims will be described in further detail.

3.1 Feature Extraction from Pros & Cons

Figure 3 shows the architecture of the feature extraction process from the pros and cons. But first we need to clarify what is meant by *feature*.

In previous works [6], the term *feature* has been used to refer to those aspects of a product that customers have expressed opinions on, also called *opinion features*. In our case, the term *feature* is used to refer to those phrases which express opinion about particular aspects of the product. In other words, [6] calls camera features things like picture quality, size, etc.; we call camera features, phrases like: great picture quality, perfect size, easy to use, etc. These are just examples of what has been called *explicit features* [9]. We also call features implicit descriptions like: fit into pocket, which refers to size. We do not make the distinction between explicit and implicit features.

In general, the content of each tag is a list of pros or cons, respectively. These pros or cons are listed and separated by delimiters and other enumeration marks. For example, in Figure 2, we can see the following pros:

```
<pros> fast, water resistant, compact </pros>
```

Here we have three pros which are separated by commas. In order to find those phrases which are probable features, a parser is used. Once all phrases or probable features are extracted, their frequency is counted and we focused on those which many people have provided opinions on. We consider frequent phrases as the ones with frequency greater or equal than three. Then, in order to keep just those which are high quality features, a manual selection process is performed. Additionally, we extend this initial set of features, exploring the infrequent features (frequency less than three), and searching for more instances of known features and increasing their frequencies. We also extend the cons using the pros in a negated way by adding not or no at the beginning. For example, if *easy to use* is a known feature in the pros, we will search its negation, *not easy to use*, in the infrequent features.

Finally, part of speech tagging is performed with the final list of features and their syntactic patterns are also obtained. For example, patterns like *JJ NN* and *JJ NN NN²* are included in the list of 2-grams and 3-grams respectively since they are quite common.

3.2 SentiWordNet

We intended to use SentiWordNet³ [4], as a way to determine the polarity of the new features extracted. We tested on the features previously extracted from the pros and cons of digital cameras, which also were manually selected. We assume that the features extracted from the pros must be positive and the features from the cons must be negative.

²JJ means adjective and NN noun

³<http://sentiwordnet.isti.cnr.it/>

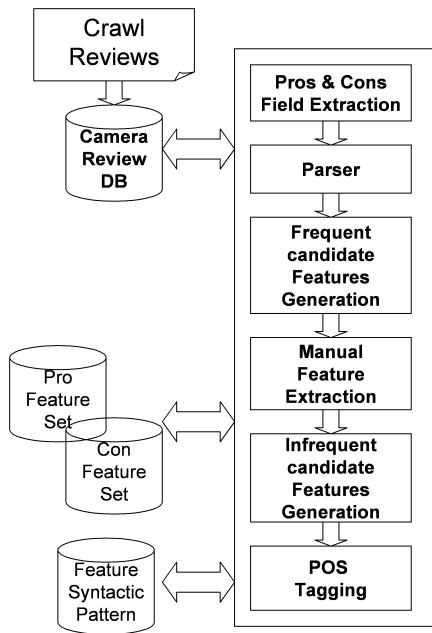


Figure 3: Feature Extraction from Pros and Cons.

Table 3 shows the results after using SentiWordNet. Please note that a considerable percentage of the features are neutral or unknown. Most of the features were not classified as either positive nor negative, which does not give any definite answer. Some examples are the following known pros, which were classified as neutral: huge lcd, large display, quick startup, quick response, compact design, long zoom.

Therefore we decided to use it just as a complementary tool, but it does not have the last word.

Pros	Size	Positive	Negative	Neutral	Unknown	Mixed
1-gram	92	35.87%	9.78%	28.26%	26.09%	0%
2-gram	347	54.18%	5.76%	30.84%	6.92%	2.31%
3-gram	171	52.05%	3.51%	38.60%	4.68%	1.17%
4-gram	35	77.14%	5.71%	17.14%	0%	0%
5-gram	7	71.43%	14.29%	14.29%	0%	0%
Cons	Size	Positive	Negative	Neutral	Unknown	Mixed
1-gram	11	36.36%	27.27%	36.36%	0%	0%
2-gram	93	8.60%	38.71%	47.31%	5.38%	0%
3-gram	93	15.05%	31.18%	48.39%	4.30%	1.08%
4-gram	20	15%	40%	40%	5%	0%
5-gram	3	0%	66.67%	33.33%	0%	0%

Table 3: Camera Feature Sentiment according to SentiWordNet.

3.3 Google-Ngrams

Two resources were constructed from the Google N-grams. The first one is a *term to term* similarity matrix which is the result of learning word associations by mining the coordination of plurals: Xs and Ys and the coordination of terms X and Y which have similar endings like: dancing and singing, connotation and implication, etc. The frequency of these patterns is used to build a list of term to term similarity, which together with Wordnet similarity gives a value to the relation, the format of this list are comma separated values with the frequency seen as follows:

40. hysterectomy [oophorectomy(91), mastectomy(91), cholecystectomy(88), ovariectomy(87), hysterotomy(78), section(43), abortion(41), comedy(14)] 41. abraham [ibrahim(96), isaac(90), noah(88), ishmael(87), joseph(87), jacob(86), benjamin(86), judah(85), graham(45)] 42. isaac [ishmael(91), abraham(90), jacob(89), benjamin(86), joseph(86), noah(86), associate(1)]

The second resource is an *adjective to adjective* matrix which is the result of learning the word associations of the form *as adjective as adjective*. In the same way, the frequency is used together with the Wordnet similarity to give a value to the relation. The format of this list is the same as the one for the term to term file.

3.4 Similarity Matrix Construction

As we have mentioned before, in order to extract new features, we use a similarity function that helps us to determine which candidate features are the most similar to the known features. Figure 4 shows the architecture overview of the building process of the similarity matrix between known features and candidate features. The input of the process is two lists of features with their POS tags and the output is the similarity matrix among all these features.

Definition (similarity matrix). Let $F_c = \{f_{c1}, f_{c2}, \dots, f_{cn}\}$ be a set of n known features for digital cameras and let $F_{p_k} = \{f_{p_k1}, f_{p_k2}, \dots, f_{p_k t}\}$ be a set of t candidate features for the product $p_k \in P = \{c, m\}$. P is the set of products, which just contains digital cameras c , and cellphones m .

Let $A_{ij} = Sim(f_{p_k i}, f_{c j})$ be a similarity matrix where $0 \leq A_{ij} \leq 100$, $1 \leq i \leq t$ and $1 \leq j \leq n$.

The similarity of each candidate feature $f_{p_k i}$ with each camera feature $f_{c j}$ is calculated with the similarity function Sim , this function is presented in Section 3.4.1.

Before we can actually compute the similarity matrix some requirements have to be fulfilled. There are three constraints:

1. The features must have the same number of words. 1-grams may only be matched with other 1-grams, etc.
2. The features have the same POS. It is only possible to match noun to noun, verb to verb, adjective to adjective, etc.
3. The features must share the same polarity or sentiment. This is determined with the help of SentiWordNet, both have to be positive, negative or neutral, etc.

3.4.1 Similarity Function

Let $f_i = \{w_1, w_2, \dots, w_l\}$ and $f_j = \{v_1, v_2, \dots, v_l\}$ be features to compare, they both contain the same number of words l according to the first constraint. We first construct the matrix S_{st} where the similarity of each word $w_s \in f_i$ with $v_t \in f_j$ is calculated.

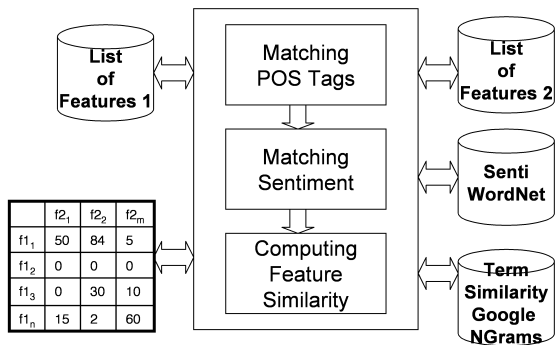


Figure 4: Similarity Matrix Construction.

$$S_{st} = \begin{cases} sim(w_s, v_t) & \text{if } POS(w_s) = POS(v_t) \\ & \text{and } SEN(w_s) = SEN(v_t) \\ -\infty & \text{if } POS(w_s) \neq POS(v_t) \\ & \text{or } SEN(w_s) \neq SEN(v_t) \end{cases}$$

where $1 \leq s, t \leq l$, $0 \leq sim(w_s, v_t) \leq 100$ this value comes from the resources described in Section 3.3. POS is the part of speech tag of the word and SEN is the sentiment of the word according to SentiWordNet.

Let $C_l^l = \{c_1, c_2, \dots, c_r\}$ be the set of all possible ways of matching f_i and f_j . Each c_k is an array of size l , and each cell contains an element t . For example, if $l = 2$ that would mean that both, f_i and f_j are features of two words (2-grams). Therefore $C_2^2 = \{c_1, c_2\}$ where $c_1 = [1, 2]$ and $c_2 = [2, 1]$. These would mean that for c_1 , w_1 matches v_1 and w_2 matches v_2 . In the same way for c_2 , w_1 matches v_2 and w_2 matches v_1 .

$$Sim(f_i, f_j) = \frac{\max \left(\sum_{\substack{1 < k < r \\ 1 < s < l}} S_{st} \quad t = c_k[s] \right)}{l}$$

For each combination c_k , the sum of the similarities term to term S_{st} is calculated. The $Sim(f_i, f_j)$ is then determined as the combination c_k which maximizes the sum of the similarities S_{st} , this value is then divided by the size of the features l .

3.4.2 Sentiment Computation

We determine sentiment in two different levels: first, the sentiment or polarity at the word level, and second, the overall sentiment of a phrase or feature level. SentiWordNet is used in both cases. The input is the term or phrase with the corresponding part of speech tags.

In order to determine the *polarity of a term*, there is an issue we have to consider; for a single word SentiWordNet might have associated many part of speech tags, and many senses. Because of this we look for all senses of the given term with the given part of speech. If there are any, we count how many of these senses are positive, negative and neutral. There are several cases:

- majority is positive then the term is called positive,
- majority is negative then the term is called negative,
- majority is neutral then the term is called neutral,
- same number of positives and negatives, the term is called neutral,
- same number of positives and neutrals, the term is called positive,
- same number of negatives and neutrals, the term is called negative,
- same number of positives, negatives and neutrals, the term is called neutral.
- unknown, if SentiWordNet does not contain it.

There are three cases to determine the overall *sentiment of a phrase*: first, the polarity of the adjective will be the overall sentiment of the phrase. If “no” or “not” come before the adjective, the polarity is inverted, if it was positive, it will become negative, if it was negative it will become positive, neutral stays the same. Second, if there is no adjective, the polarity of the adverb will be the overall sentiment of the phrase. Finally, if there is no adjective nor adverb, the polarity of a phrase will be given by its individual terms. At least the ones which are contained in SentiWordNet. We count how many positive, negative and neutral there are and we will have several cases:

- majority is positive then the phrase is called positive,
- majority is negative then the phrase is called negative,
- majority is neutral then the phrase is called neutral,
- mixed if none is outstanding,
- unknown, if SentiWordNet does not contain any.

3.5 POS Tagging

The tagger used in this work is TreeTagger⁴ [15]. Using this tagger we parse the content of each field (pros or cons), dividing it into sentences and yielding the part of speech tag for each word together with its base form. TreeTagger also includes a *chunker* which identifies groups, like noun chunks or verb chunks. The output of the TreeTagger chunker looks like this:

<NC> long JJ battery NN life NN </NC> . SENT

It gives a tag for each word, and the chunks are pointed out using XML tags, in the example above we are seeing a noun chunk.

⁴<http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>

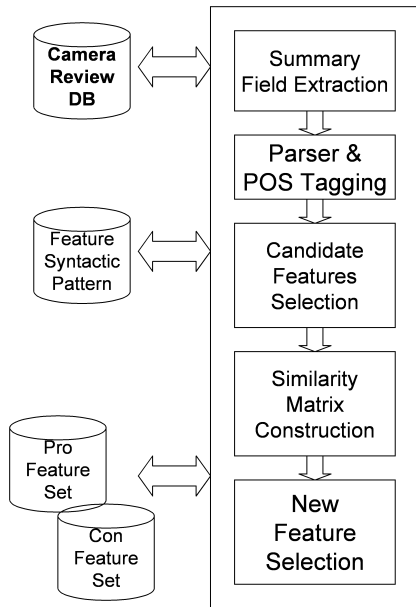


Figure 5: Feature extraction from summaries.

3.6 Summary Feature Extraction

Figure 5 shows the process of extracting new features from the summary fields. The input of this process is the reviews downloaded for a product, the format of which is shown in Figure 2. The output is the list of new extracted features. Given the input, the first step is to extract the xml tags that refers to the summaries of all reviews in the training dataset. These summaries are then passed through a parser/chunker to extract the sentences, and determine their syntactic structure, based on the list of syntactic patterns previously collected for cameras similar to the phrases with a known syntactic pattern are selected as new candidate features. The matching is done in size decreasing order, first 5-gram, then 4-gram and so on. The candidate features list is reduced, taking into account the polarity of the features. If it is the case that we are looking for pros, the features that have a negative sentiment are not taken into account, and the analogous step is done for cons. Using the similarity function described in section 3.4, the similarity between the candidate new features and the known features is computed. Finally a threshold is set, and just the features which are over the threshold will be accepted as definite new features.

4. CROSS TRAINING PRODUCT FEATURES

The goal of this investigation is to understand the relation between features of crossproducts, i.e. using features from one product to find features in other products. In particular, we are using the features of digital cameras, to extract cellphone features. In this paper, we only focus on extracting the cellphone features from the pros and cons fields. The use of the summary fields will be discussed in a future paper.

4.1 Extracting Cellphone Pros and Cons

Figure 6 shows the cellphone feature extraction process which is done in several steps. The process is quite similar to the summary feature extraction. The input is the cellphone re-

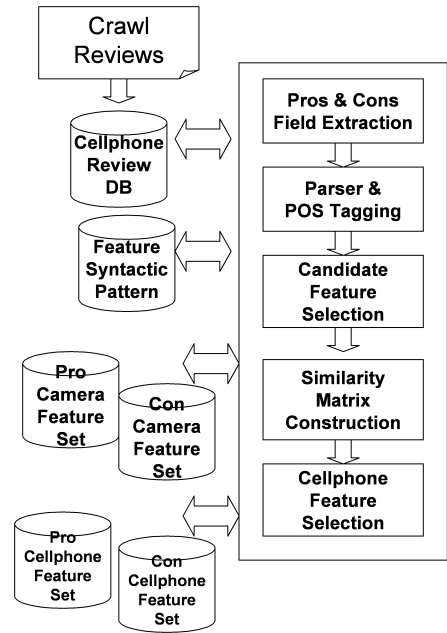


Figure 6: Feature extraction from cellphone pros and cons.

Pros	Features	Size	Cons	Features	Size
1-gram	92	7 kb	1-gram	11	1 kb
2-gram	347	50 kb	2-gram	93	9 kb
3-gram	171	30 kb	3-gram	93	8 kb
4-gram	35	3 kb	4-gram	20	2 kb
5-gram	7	1 kb	5-gram	3	1 kb
6-gram	2	1 kb	6-gram	1	1 kb

Table 4: Initial set of features for digital cameras.

views crawled from the CNet site, which follows the same format that the one shown in Figure 2. The outputs are the lists of extracted features (pros and cons). The first step is, given the input, to extract the xml tags that refers to the pros and cons fields of all cellphone reviews in the test dataset. From now on, the process is exactly the same as the summary feature extraction.

5. EVALUATION

Table 4 shows the number of extracted features from the pros and cons of digital cameras. The top most selected camera features can be seen in Table 5.

Unigrams	Bigrams	Trigrams	Fourgrams
Most Frequent Pros			
good	image stabilization	easy to use	very easy to use
great	great picture	ease of use	easy to use menu
small	small size	long battery life	very good picture quality
light	manual control	great picture quality	easy to use control
fast	great camera	lot of feature	build like a tank
compact	good picture	great image quality	lot of manual control
thin	movie mode	great battery life	easy to navigate menu
easy	lcd screen	excellent picture quality	easy to use software
Most Frequent Cons			
slow	low light	no image stabilization	noise at high iso
big	too small	no optical zoom	only 3x optical zoom
noise	shutter lag	a little slow	noisy at high iso
expensive	manual focus	short battery life	grainy in low light
blurry	weak flash	no battery meter	no battery life indicator
heavy	small lcd	no manual control	no optical view finder
bulky	manual control	no optical viewfinder	poor low light performance

Table 5: Top most frequent features for digital cameras.

5.1 Quality of summary-derived Features

After the similarity matrix is built, an acceptance criterion or threshold needs to be set to assure the quality of the features extracted. Table 6 shows how the new features extracted are distributed when different thresholds are set. It can be seen that there are big jumps from interval to interval, which does not tell us anything meaningful about which threshold we should pick.

Pros	Similarity >50	Similarity >60	Similarity >70	Similarity > 80	Similarity > 90
1-gram	339	243	124	55	11
2-gram	1688	1336	945	506	155
3-gram	2698	1537	718	356	100
4-gram	366	256	150	53	19
5-gram	35	25	15	5	4
6-gram	1	1	0	0	0
Cons	Similarity >50	Similarity >60	Similarity >70	Similarity > 80	Similarity > 90
1-gram	0	0	1	0	0
2-gram	611	428	291	148	34
3-gram	556	250	47	29	6
4-gram	35	31	23	5	1
5-gram	1	1	1	0	0
6-gram	0	0	0	0	0

Table 6: Summary Feature Distribution according to Similarity.

To evaluate the quality of the automatically selected features a manual inspection is performed. Taking into account that most of the features are concentrated in the 2-grams and 3-grams, the evaluation was done in just these last two groups. But first some further filters were applied. SentiWordNet was used to reduce the number of retrieved features; if we are looking for pros, just the features explicitly qualified as positive are accepted, the same for cons, just the features qualified as negative are accepted. After manually evaluating it was noticed that most of the eliminated features had a small frequency, such as the syntactic patterns shown in Table 7.

	Pros		Cons
3-grams	NN IN NN NN TO VB	3-grams	DT NN NN DT JJ NN
2-grams	NN NN JJ NN	2-grams	NN NN DT NN

Table 7: Most common syntactic patterns of eliminated features.

The lists of candidate features (with strict and weak SentiWordNet) were filtered, those which had any of the given patterns as a structure and a frequency equal to 1, were eliminated from the lists. The results of this process are shown in Figures [7, 8, 9, 10].

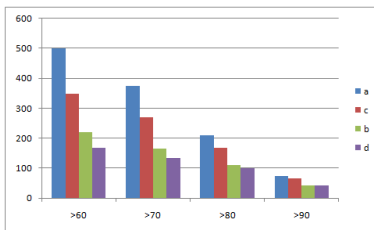


Figure 7: 2-grams: pro features from camera summaries. a)Sim+Weak SWN, b)Sim+Strict SWN, c)Sim+Weak SWN+Manual Evaluation, d) Sim+Strict SWN+Manual Evaluation. The X-axis are sim values, Y-axis are number of features.

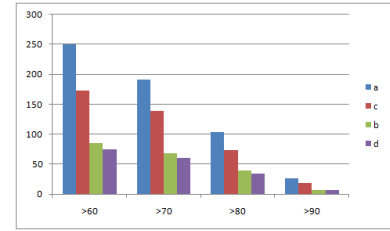


Figure 8: 2-grams: con features from camera summaries. a)Sim+Weak SWN, b)Sim+Strict SWN, c)Sim+Weak SWN+Manual Evaluation, d) Sim+Strict SWN+Manual Evaluation. The X-axis are sim values, Y-axis are number of features.

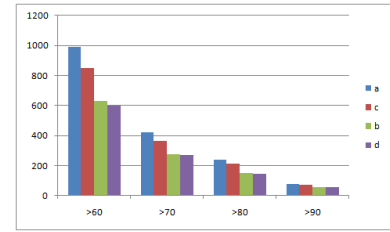


Figure 9: 3-grams: pro features from camera summaries. a)Sim+Weak SWN, b)Sim+Strict SWN, c)Sim+Weak SWN+Manual Evaluation, d) Sim+Strict SWN+Manual Evaluation. The X-axis are sim values, Y-axis are number of features.

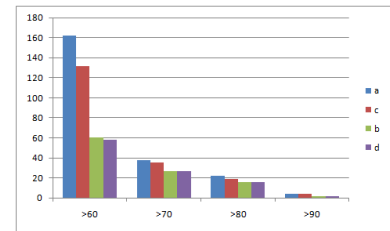


Figure 10: 3-grams: con features from camera summaries. a)Sim+Weak SWN, b)Sim+Strict SWN, c)Sim+Weak SWN+Manual Evaluation, d) Sim+Strict SWN+Manual Evaluation. The X-axis are sim values, Y-axis are number of features.

The figures show that in all cases the column b and d are always below c, meaning that SentiWordNet eliminates many of the good features that were already obtained by the similarity function. It also shows the method Similarity + Strict SentiWordNet still has some noise. Both methods *Similarity + Strict SentiWordNet* and *Similarity + Weak SentiWordNet* with the threshold set to 60% have an accuracy around 75%. There is a considerable portion of noisy features, therefore noise analysis needs to be performed to understand where the noise is coming from and if it can be avoided.

5.1.1 Cross-product Feature Coverage

To evaluate the quality of the automatically extracted features from the pros and cons of cellphones, these are compared with the list of manually selected features (which comes from the list of frequent features in pros and cons). Since the features are mainly concentrated in the 2-grams and 3-grams, the evaluation was done just in these last two groups. For the automatically extracted features we use the ones with similarity greater than 70. Table 8 shows the results of comparing the clean frequent features with the automatically extracted ones. In the top of the table together with the headers pros and cons we can find the sizes of the set of clean features (pros and cons for bigrams and trigrams). In the table we can see pairs of numbers. The first is the number of features in the set of automatically extracted features, the second number in parenthesis means the number of features that both sets (automatically extracted and clean) share.

	Bigrams		Trigrams	
	Pros(671)	Cons(233)	Pros(315)	Cons (181)
>70	1300(295)	284(35)	1052(94)	146(24)
>80	750(192)	131(21)	521(59)	68(9)
>90	214(70)	36(4)	120(16)	18(5)

Table 8: Cross-product feature intersection

In this experiment a low accuracy is also shown. The next step would be to perform a noise analysis to detect where the noisy features are coming from and give

6. CONCLUSION

In this work, we have explored a novel feature similarity method, between the features extracted from user opinions on electronic gadgets. A corpus was developed to evaluate our method. We found that using SentiWordNet in a strict way gives good quality features but with a very small set of features. The manual evaluation (bar number c) showed that we have missed a considerable number of good features using SentiWordNet strictly. We found that using SentiWordNet *relaxly* helps to grasp more good pictures than using it *strictly* although it introduces some noise.

7. REFERENCES

- [1] S. Bethard, H. Yu, A. Thornton, V. Hatzivassiloglou, and D. Jurafsky. Automatic extraction of opinion propositions and their holders. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text*, 2004.
- [2] Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 355–362, Morristown, NJ, USA, 2005. Association for Computational Linguistics.
- [3] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 519–528, New York, NY, USA, 2003. ACM.
- [4] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, pages 417–422, 2006.
- [5] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM.
- [6] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of AAAI*, pages 755–760, 2004.
- [7] S.-M. Kim and E. Hovy. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 483–490, Morristown, NJ, USA, 2006. Association for Computational Linguistics.
- [8] B. Liu. *Sentiment Analysis and Subjectivity*. CRC Press, Taylor and Francis Group, Boca Raton, FL, USA, second edition, 2010.
- [9] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 342–351, New York, NY, USA, 2005. ACM.
- [10] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, pages 115–124, 2005.
- [11] B. Pang and L. Lee. *Opinion Mining and Sentiment Analysis*. Now Publishers Inc, July 2008.
- [12] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.
- [13] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 339–346, Morristown, NJ, USA, 2005. Association for Computational Linguistics.
- [14] C. Scaffidi, K. Bierhoff, E. Chang, M. Felker, H. Ng, and C. Jin. Red opal: product-feature scoring from reviews. In *EC '07: Proceedings of the 8th ACM conference on Electronic commerce*, pages 182–191, New York, NY, USA, 2007. ACM.
- [15] H. Schmid. Probabilistic part-of-speech tagging using decision trees, 1994.