SEMANTIC BASED UNSUPERVISED LEARNING APPROACH FOR FEATURES EXTRACTION FROM TEXT REVIEWS

U. Faheem[1], E.N.G. Weng[2]
[1] Faculty of Cognitive Sciences & Human Development, Universiti Malaysia Sarawak
Kota Samarahan, 94300, Malaysia
faheem_bcs@yahoo.com
[2] Faculty of Cognitive Sciences & Human Development, Universiti Malaysia Sarawak
Kota Samarahan, 94300, Malaysia
nggiapweng@yahoo.com

ABSTRACT: The existing research shows that reviews are important for both customers and companies. To extract and analyze customer reviews from Internet, an efficient system is needed. The existing literature shows that features and syntactic patterns have a great role in the extraction of specific features, sentiments and information from text. The focus of this paper is the extraction of sentimental features from product reviews. A domain independent semantic based unsupervised technique is proposed for automatic extraction of features from reviews. This technique exploits syntactic patterns and semantic relations by analyzing them to identify refined explicit features. Experiments on different products reviews are conducted and compared the results with existing methods. The results show that the proposed approach outperforms the existing approaches.

Keywords: *Opinion mining, NLP, features refinement, features purification, sequential patterns.*

1. INTRODUCTION

Opinions are important for organizations as well as individuals due to its impact on decisions and policy making. People's thoughts are significant for one's guidance. Today, huge amount of information is available on Internet in the form of social networks, e-commerce sites, online news groups, comment boxes, online forums, blogs and billions of websites. Businesses spend money and time for finding customers' opinions regarding their products and services for satisfaction by taking help from consultants and conducting surveys [1,2]. Similarly researchers are concerned in people's opinions about products, services, topics and events for determining the best choices [3]. Due to the availability of these online contents, the opinion mining (OM) and sentiment analysis (SA) has become a very hot research topic recently. It extracts and analyzes opinions based on features from text or reviews for predictions and decision making using data mining, natural language processing and machine learning techniques.

OM and SA tools process reviews, extract features and aggregate opinions. An efficient system for aggregation and summarization is needed for making possible online analytical processing [4]. Researchers try to make computer able to recognize, understand, generate emotions and take intelligent decisions. Feature mining is an important task for opinion mining and sentiment analysis as it provides base for opinion understanding [5, 6]. In this paper our approach is features extraction i.e. to identify and extract features from each sentence in a review of a particular object, topic or event upon which the reviewers have expressed opinions. A feature-based SA and OM approach has been used by [6, 7, 9-11].

We have developed an efficient domain independent semantic based unsupervised learning algorithm for the extraction of features from text. This algorithm exploits potential syntactic patterns for the identification of linguistic features. Two most popular unsupervised approaches for opinion targets extraction are association mining [6, 11, 12] and likelihood ratio test (LRT) [7, 9]. Association mining based products features extraction was used by Hu and Liu [6]. Wei *et al.* [11] proposed product features extraction methodology through association mining with semantic based pruning. Yi *et al.* [7] used LRT for product features extraction. LRT based product features extraction with subsequent similarity was used again by Ferreira *et al.* [9]. This technique checks the frequency distribution of terms to identify relevance of target features to the topic. The work of Ferreira et al. [9] shows that LRT outperforms association rule mining approach for features selection. Hence in this paper LRT has been used for the selection of frequently occurred features. The technique has been enhanced by adding semantic based relevancy of features by using WordNet, as LRT is based on threshold which depends on frequency of terms. Perfect value for threshold cannot be identified; that is why features having low frequencies are misclassified. We have introduced an approach called semantic based LRT (SLRT) based on syntactic patterns along with semantic relations for the extraction of features from text, which outperform the existing approaches.

2. Proposed Work

The proposed work presents algorithm for features extraction through targets from products reviews for OM & SA task. Opinion target is the user's concern about which opinion holder expresses opinion. When we express opinion, it will be about an entity or about an attribute of an entity. The entity or the attribute is called target of the opinion holder. Every opinionated sentence in document comprises target about which opinion is expressed. The identification of opinionated expressions through opinion targets is the key for features extraction. The proposed approach has three main steps, shown in Figure 1.

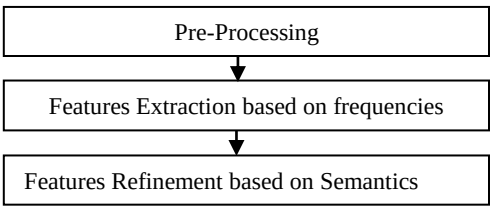

Figure. 1 over all Process of Features Extraction

2.1 Preprocessing

The natural language text consists of number of unnecessary terms. Therefore text cleansing is necessary before input into actual process. The most commonly used text pre-processing techniques are POS tagging, Phrase Chunking. Furthermore we have proposed pruning algorithm in order to stem and group terms with semantic relations.

a. Parts of Speech Tagging (POS)

POS is crucial pre-processing steps related to NLP. Part-of-Speech Tagging is used to assign corresponding category to each word based on its definition and context in a given text. Actually the proposed approach is based on sequential patterns of lexical categories of words/terms, therefore POS is essential before further processing. We use POS both for patterns extraction and for NPs and VBs identification for target features identifications. For implementation of this step we use state of the art POS software, the Stanford POS tagger [8], which is frequently used in natural language processing (NLP) practices. The pre-processing is shown in Figure 2.

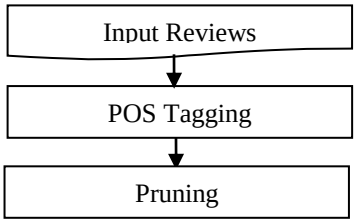

Figure. 2 Pre-Processing steps

b. Pruning

In this step we convert all plural to singular e.g. cameras/NNS to camera/NN, families/NNS to family/NN, Pictures/NNS to Picture/NN etc. This step is important for synonyms extraction using WordNet. The objective of this step is to reduce the size of candidate features.

2.2 Target Features Extraction

This section describes the process of extraction of target features from the reviews. This process consists of two main steps: Candidate features extraction and relevance scoring as explained below.

a. Candidate Features Extraction

In candidate features (CF) extraction step, the extraction of noun and verb phrases as candidate features is done. We extract only noun phrases from documents and apply feature selection algorithms. In order to extract these phrases we use regular expressions. The nouns and verbs are already identified in POS tagging steps. We just extract NPs and VPs using the following regular expressions.

Noun phrases

- *NP → JJ* NN+ CD**
- *JJ →Adjective [having no positive or negative polarity]*
- *NN → Noun*
- *CD → Digits*

Verb Phrases

- *VP → (VB+)**
- *VB → Verbs*

b. Relevance Scoring

The relevance scoring has two broad categories i.e. relevance scoring based on distributional similarity and relevance scoring based on dependency on pre-existing knowledge resources like thesauruses, ontology or encyclopedias. This paper exploits the unsupervised distributional similarity method for features extraction; the LRT, discussed in section 1.

Likelihood Ratio Test: Let D_+ represents texts related to topic *T* and D_- represents texts not related to the topic *T*. The *BNPs* in D_+ are candidate features which will be classified as topic relevant or irrelevant using LRT as: if the likelihood score of *BNP* satisfies the predefined threshold value then *BNP* is considered as target feature. Here,

$$r_1 = \frac{n_{11}}{n_{11}+n_{12}} \quad (1), \quad r_2 = \frac{n_{21}}{n_{21}+n_{22}} \quad (2)$$

$$r = \frac{n_{11}+n_{21}}{n_{11}+n_{12}+n_{21}+n_{22}} \quad (3)$$

where r_1 is the ratios of relevancy of the *BNP* to topic and r_2 is the ratios of relevancy of the *BNP* to non-topic. r is the combined ratio. n_{11}, n_{12}, n_{21} and n_{22} are shown in Table 1.

Table 1: Contingency Table

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

$$lr = (n_{11}+n_{21})\log(r) + (n_{12}+n_{22})\log(1-r) - n_{11}\log(r_1) - n_{12}\log(1-r_1) - n_{21}\log(r_2) - n_{22}\log(1-r_2) \quad (4)$$

lr is the normalized ratio with *log*. λ is the likelihood ratio. The log likelihood ratio $-2\log\lambda$ is calculated as:

$$-2\log\lambda = \begin{cases} -2*lr, & if_1\ r_2 < r_1 \\ 0 & if\ \ r_2 \geq r_1 \end{cases} \quad (5)$$

The likelihood ratio is directly proportional to the value of $-2\log\lambda$. If the value of $-2\log\lambda$ increases the relevancy of *BNP* to the opinion targets also increases. For each *BNP*, compute the likelihood score $-2\log\lambda$ as defined in equation 5. Then sort *BNP* in decreasing order of their likelihood score. Feature terms are all *BNPs* whose likelihood ratio satisfies a pre-defined confidence level. Simply only the top *n BNP*'s can be selected.

c. Key Features Grouping

Sequential patterns of lexical terms are used for the extraction of key features groups (KFGs) from the input documents. Initially CF is extracted and ranked based on frequencies. Then the proposed method extracts KF from CF and builds key features groups (KFG) from KF based on semantic similarities and relatedness. The KFGs are further updated through key features group matrix (KFGM) and

WordNet to get final groups of related target features (TFs). The process is shown in the Figure 3.

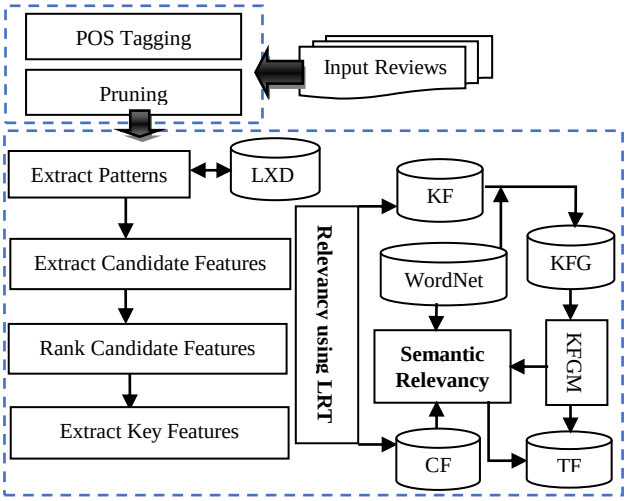

Figure 3 Pattern-Based Key Features Groups (KFG) Extraction Process

c.1 Key Features Extraction

KF extraction process is the core task of our proposed technique. KF are those which are extracted from CFs and CF are those which are extracted through our pre-defined patterns. After through investigations and a lot of experiments, some patterns were found which showed strong indications towards object names or features. The noun phases in the patterns are proposed as key features. In order to extract KF we use restricted Sequential Patterns (SP) of terms. In each sentence our algorithm searches the following two types of patterns.

NPVBJJ

NPVBRBJJ

- *Pattern →NPVBJJ*
- *Pattern →NPVBRBJJ*
- *NP →JJ*NN+*
- *JJ →Adjective*
- *NN → Noun*
- *VB → Verb*
- *RB → Adverb*
- *NN → Noun*
- *VB → Verb*

c.2 Candidate Features Ranking

This step is used to prepare sorted list of CFs based on their frequencies. The aim of creating ranked list of NPs is to hypothesize that an NP with highest frequency has the probability to be target feature (TF). Based on this scenario the list is ordered in descending order.

c.3 Create Key Features Groups

In this step we create groups of key features in the document. We call the group of such features as KFG. Based on the key feature we identify them as Target Objects and Target Features in the document. The input to this step is the ranked list of feature obtained in the previous step. In order to create group of key features we use Hyponym relation from WordNet. Hyponymy is a relation between meanings, so it holds among synsets. For example the synsets (digital, camera) is a hyponymy of the synsets (camera) in WordNet. The following algorithm creates key features groups.

- *Given a sorted list of key features L as input, the following algorithm creates Key Features Groups (KFG).*
- *Input term: Key Features List(L)*
- *Output term: Key Features Group(G)*
- *Body*
- *For each Feature F in L*
  - *if F is unmarked then*
  - *Create Group Gi={F}*
  - *set F as header of Gi*
  - *Mark F in L*
  - *Find Hyponyms Hp of F in Word Net*
  - *Create List of Hyponyms: Li=Null*
- *For Each Hi in Hp*
  - *For each unmarked terms T in L*
    - *If T is equivalent to Hi then*
    - *Put T in the list: Li=Li ∪ T*
    - *Mark T*
    - *Put related list in Group Gi: Gi=Gi ∪ Li*
  - *Next*
- *Next*
- *Next*

c.4 Create Features Groups

In this step corresponding terms are extracted from CF based on KFG. Initially NPs and VBs are extracted as CF. The proposed algorithm checks semantic relation of each rejected CF by LRT shown in Figure 3 to KFG using WordNet. There are three possibilities for each CF. Either it may be exactly same as term in a KFG, or it may be semantically related to a term in one or more than one KFG or it may not be related to any KFG. In the first case no further computation is needed and is directly included in the KFG to which it is related. In the second case we use cosine similarity measure through KFGM matrices. In the third case we add a new group to the matrix in order to avoid removal of infrequent features. The detail about KFGM matrix is given in subsequent sections.

c.4 Cosine Similarity

Cosine similarity or cosine angle distance is used in this research for finding similarity amongst features vectors. In comparison with Euclidean distance the cosine angle distance is more suitable in documents retrieval [12].

c.5 Create Key Feature Groups Matrix

Initially, matrix is created from KFG data. Matrix is created from the dictionary of words included in each KFG. Each row includes a relation about a single KFG while the intersection point of matrix provides frequency of a word in the KFG. Through KFG matrix, a group of CF is predicted i.e. in which KFG a CF may be best fitted. After classifying all the CFs of input documents, yields a matrix of key features groups called KFGM. From KFGM, target features (TF) are investigated using Meronym relations in WordNet. KFGs and CFs are needed as input to create and update KFGM. Each CF in the CF list is checked and the related KFG is predicted for that specific CF by finding probability for that CF using cosine similarity function. Cosine function

is used to find similarity between features. Finally the CFs are added to their respective KFGs and the KFGs are added to KFGM. Template of KFGM matrix is shown in Figure 4.

- *Create-Features-Groups: Given a Key Features Groups (KFG) and List of Candidate Features (CF) as input, the following algorithm creates a set of Features Groups (FG).*
- *Input term: Key Features Groups(KFG), Candidate Features(CF)*
- *Output term: Key Features Groups Matrix (KFGM)*
- *Body*
- *For each Candidate Feature f in CF*
  - *Predict Key Feature Group for f: KFG=Predict-KFG(f)*
  - *update KFG: KFG=KFG ∪ f*
  - *KFGM={KFGM} ∪ {KFG}*
- *Next*
- *Predict-KFG: Given a Candidate Feature CF and List of Key Features Groups (KFG), the following algorithm Predicts KFG of the CF.*
- *Input: Candidate Feature(CF), KFGM Matrix*
- *Output: KFG*
- *Body*
- *For Each KFG in KFGM Matrix*
  - *Find Cosine Probability p for CF: Cos(CF)=*
  - *find KFG with maximum probability (p)*
  - *if p>0 then*
    - *Return KFG*
    - *else*
    - *Add New KFG*
  - *End*
- *Next*

$$KFGM_{|D|\times F} = \begin{bmatrix} FG & W_1 & W_2 & \cdots & W_i & \cdots & W_n \\ H_1 & f(w_1) & f(w_2) & \cdots & f(w_i) & \cdots & f(w_n) \\ H_2 & f(w_1) & f(w_2) & \cdots & f(w_i) & \cdots & f(w_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ H_i & f(w_1) & f(w_2) & \cdots & f(w_i) & \cdots & f(w_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ H_n & f(w_1) & f(w_2) & \cdots & f(w_i) & \cdots & f(w_n) \end{bmatrix}$$

Figure 4 Sample KFGM Matrix in the form of table

c.6 Identify Target Object & Target Features

Finally, target objects (TO) and target features (TF) are identified in the FG using meronym relationship. Meronymy is a relation between a whole and its parts. The meronym relations are taxonomy of objects and their features like part-of, member-of, substance-of etc. Fortunately WordNet provides this type of taxonomy relations the meronyms relations. Therefore, TO and TF are identified through WordNet meronym relation in each group based on the matching sense. The algorithm of identifying TO or TF is shown below.

- *Identify-Targets: Given a Key Features Groups Matrix (KFGM) the following algorithm identifies Target Object (TO) and Target Features (TF).*
- *Input term: KFGM*
- *Output term: List of Target Objects (LTO) and List of Target Features (LTF)*
- *Body*
- *For Each Header H in KFGM*
  - *Find Meronyms Tree MT*
  - *If found MT then*
  - *Find Level L for H in MT*
  - *if H is on the Root then*
  - *LTO={LTO} ∪ H*
  - *Else*
  - *for each Feature F in H row of KFGM*
    - *Find Meronyms Tree MT*
    - *If found MT then*
    - *Find Level L for H in MT*
    - *if H is on the Root then*
    - *LTO={LTO} ∪ {F}*
  - *Next*
- *Next*
  - *LTO= {LTO} ∪ {F} ∪{F}*

2.3 Tools

There are two main steps in the implementation of this methodology i.e. extraction of patterns, and features using the proposed patterns and opinion lexicon. For pattern extraction and verification, TextStat 3.0 [13] has been used and the proposed algorithm for features extraction has been implemented. It takes input extracted through TextStat 3.0 and checks the subjectivity of the adjective in each input patterns to determine the opinion hood of the expressions. The module then produces the list of features from the opinionated expressions. Finally it checks the extracted features with the list of manually annotated features in the corpus and calculates the evaluation matrices.

3. RESULTS

In this paper, the patterns based approach has been exploited. For candidate features selection some researchers have used heuristics, based on base noun phrases [7, 9]. From [9] research work the definite base noun (dBNP) phrases have been selected as dBNP provides better results than BNP and bBNP. As given in Sections 2.2.a and 2.2.c.1 the candidate features are extracted through the patterns and then the likelihood is calculated to select targets based on relevance scoring. If for a given candidate feature the value of $-2\log\lambda > 0$, then it is considered as feature. For each of the prescribed setup; true positive, true negative, false positive and false negatives are calculated. Based on those values precision, recall and f-score is calculated. Finally semantic based relevancy process takes place, hence called semantic based LRT (SLRT). Based on comparative results with existing approaches it is found that the proposed approach perform better than existing pattern based approaches. The results show that average frequency and Recall scores are comparatively better than Ferreira *et al*. [9].

3.1 Datasets

The proposed approach is using standard data sets of the five product reviews collected by Hu and Liu [6]. The data sets are manually annotated by [6] and [9]. These datasets have been extensively reported in number of research articles for comparative analysis of product features extraction and opinion summarization. The summary of the dataset is given in Table 3. We have compared our results with [9] as proved from literature that [9] provides better results than [6].

Table 2: Manually annotated datasets by [6] & [9]

Datasets	Total Sentences	Manually tagged features by [6]		Manually tagged features by [9]	
		Distinct Features	Total Features	Distinct Features	Total Features
APEX	738	110	347	166	519
Canon	594	100	257	161	594
Creative	1716	180	736	231	1031
Nikon	346	74	185	120	340
Nokia	546	109	310	140	470

Table 3: Comparison of the results of SLRT with [9] in terms of Precision, Recall and F-score

Datasets	Method	Precision	Recall	F-Score
Apex	LRT [9]	92.04%	51.35%	65.92%
	SLRT	91.83%	75.08%	82.61%
Canon	LRT [9]	91.50%	51.79%	66.14%
	SLRT	86.25%	65.78%	74.63%
Creative	LRT [9]	92.51%	57.54%	70.95%
	SLRT	87.99%	68.56%	77.06%
Nikon	LRT [9]	90.65%	51.35%	65.71%
	SLRT	87.89%	73.08%	79.80%
Nokia	LRT [9]	91.85%	56.35%	69.86%
	SLRT	89.95%	74.85%	81.70%

Table 4: Comparison of Average Precision, Recall and F-score

Method	Measure	Average Score
LRT [9]	Precision	91.71%
	Recall	53.68%
	F-Score	67.72%
SLRT	Precision	88.78%
	Recall	71.47 %
	F-Score	79.19%

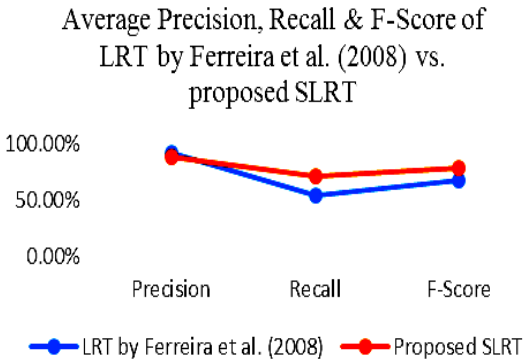

Figure 5 Comparison of Average Precision, Recall and F-score

4. CONCLUSIONS

In this paper, NLP tools to sentiment analysis practices are applied and sentiment pattern based features extraction methodology is presented to extract features and categories. The features extraction algorithm successfully extracts features through targets from online reviews. The proposed methodology uses various algorithms to solve issues related to the problem. The product reviews has been used for experimentation. NLP research shows that the methodology can be improved by applying full parsing to provide better sentence structure analysis. In future the manual validation will be implied with more effective anaphora resolution.

REFERENCES

[1] Bo Pang, Lillian Lee. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, Vol. **2**, Nos. 1–2 2008.

[2] Roberto Basili, D. C. (2011). Opinion Mining. WM&R.

[3] Hu, Bing Liu. Mining Opinion Features in Customer Reviews, Mining. *American Association for Artificial Intelligence*, 2004.

[4] Tsytsarau, M., & Palpanas, T. (2012). Survey on mining subjective data on the web. Data Mining and Knowledge Discovery, 24(3), 478-514.

[5] Bing Liu. The utility of linguistic rules in opinion mining. *international ACM SIGIR conference on Research and development in information retrieval* 2007.

[6] Hu, M and Liu, B. (2004). Mining and Summarizing Customer Reviews. *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD‟04), 2004.

[7] Yi, J., Nasukawa, T., Bunescu, R., & Niblack, W. (2003, November). Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In Third IEEE International Conference on Data Mining, 2003. ICDM 2003. (pp. 427-434).

[8] Toutanova K, Klein D, Manning C, & Singer Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. North American Association for Computational Linguistics (NAACL). p 173-180.

[9] Ferreira, L., Jakob, N., & Gurevych, I. (2008, August). A comparative study of feature extraction algorithms in customer reviews. In Semantic Computing, 2008 IEEE International Conference (pp. 144-151). Santa Clara, CA:IEEE.

[10] Chen, H. H., Lin, M. S., & Wei, Y. C. (2006, July). Novel association measures using web search with double checking. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (pp. 1009-1016). Sydney, Australia: Association for Computational Linguistics.

[11] Wei, C.P., Chen, Y.M., Yang, C.S., & Yang, C. C. (2010). Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews. Information Systems and E-Business Management, 8(2), 149-167.

[12] Qian, G., Sural, S., and Pramanik, S. (2000). A comparative analysis of two distance measures in color image database. Proceedings of international Conference on Image Processing. Vol. 1, pp. I-401). IEEE

[13] Diniz L. 2005. Comparative Review: TextStat 2.5, ANTCONC 3.0, and Compleat Lexical Tutor 4.0. *Language Learning & Technology*, *9*(3), 22-27.