6. ADJECTIVE BASED GRAMMAR RULES FOR SEMANTIC MODEL

6.1 INTRODUCTION

It is the hypothesis of this work that adjectives are the key semantic structure for classifying the genre of movies. For an example, users generally express their sentiment about a movie they watch, using words like நல்ல (good), எகட்ட (bad), and ஆச்சர்யமாக (amazing) etc.

Therefore, this chapter explains how to identify the sentiment of adjectives in order to classify the tweets. The contribution of this Chapter as follows.

- Section 6.2 presents the different types of adjectives in Tamil and the rules developed for adjectives in relation to sentiment analysis is discussed in section 6.3
- Section 6.4 describes the experimental setup and flow diagram
- Section 6.5 presents the algorithm to categorize the tweets using the patterns of adjectives
- The performance of grammar rules based on adjectives and negations are discussed in section 6.6

6.2 ADJECTIVES IN TAMIL

Most of the traditional Tamil grammars discuss only nouns and verbs. It suggests that the traditional grammarians or linguists have not considered adjectives as separate categories in Tamil. However, some of them have treated adjectives as relative participial forms of verbs (குறிப்பு பெயரச்சம் ‘kuRippup peyareccam’) and relative participial forms of finite verbs (தெரிந்தைப் பெயரச்சம் ‘terindilaip peyareccam’).

Modern Tamil language has some important classes for adjectives. Adjectives and adverbs are syntactically recognized categories in Tamil [84]. The main reason
for choosing adjective rules is that adjectives do not inflate. However, adjectives are formulated from certain nouns and verbs by adding suffix terms. The adjectives are identified by its function in a sentence construction. For an example, the most commonly used adjective suffixes are ஆன (aana) and ம் (a).

In this chapter, major efforts and time have been devoted to adapt the grammar rules for adjective words/phrase in Tamil that are focused on Tamil tweets classification. Adjective rules are a unique feature of this research that can classify most of the tweets in the corpus. The adjectives and their semantic relations tend to be domain specific; therefore in this research corpus-based approach is used to compute the semantic structure of adjectives in the Twitter.

6.3 RULES FOR ADJECTIVES

Adjectives are used to describe or quantify an object about nouns. The occurrences and function of adjectives differ in different languages. In modern Tamil, most of the times adjectives are written immediately before the noun. In this method, a pattern of adjectives is identified in relation to Tamil movie tweets. These adjective patterns have a direct relation with the particular domain specific posts, and thus rules have been developed in this context to find a sentiment to the particular user posts [91]. Adjectives are also used to express the strength of user sentiments in the form of intensifiers.

Table 6.1 Types of adjectives in modern Tamil

<table>
<thead>
<tr>
<th>Type</th>
<th>Markers</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Adjective</td>
<td>Suffix – ஆ (a)</td>
<td>பிபிப் (Periya), சியியிய (Ciriya)</td>
</tr>
<tr>
<td>Derived Adjective</td>
<td>Suffix – ஆன (aana)</td>
<td>அழகான (Azhagaana)</td>
</tr>
<tr>
<td></td>
<td>Suffix – பாது (aavatu)</td>
<td>நான்கியதுபே (Naankaavatu paTi)</td>
</tr>
</tbody>
</table>
In Tamil, the adjectives may be classified into a simple adjective and derived adjective based on their syntactical structure. While simple adjectives are not derived from any other grammatical category, derived adjectives are derived mostly from nouns. This is given in Table 6.1.

An example of each type of adjective is explained in the following set of tweets. In the below examples, tweet, tokens, POS patterns, the polarity of the tweet is shown. The parse tree structure is also given to represent the dependency between POS patterns or tokens and finally, the algorithm to compute the polarity for each negation type is given.

6.3.1 Example for Simple Adjective

# வெரம் ஒரு நல்ல படம்
(# Veeram oru nalla padam)
(English: #Veeram is a good movie)

<table>
<thead>
<tr>
<th>Tweet</th>
<th># வெரம் ஒரு நல்ல படம் (# Veeram oru nalla padam)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokens</td>
<td>ஒரு (oru)</td>
</tr>
<tr>
<td></td>
<td>நல்ல (nalla)</td>
</tr>
<tr>
<td></td>
<td>படம் (padam)</td>
</tr>
<tr>
<td>Parts</td>
<td>Noun</td>
</tr>
<tr>
<td></td>
<td>Adjective</td>
</tr>
<tr>
<td></td>
<td>Noun</td>
</tr>
<tr>
<td>Polarity</td>
<td>Positive</td>
</tr>
</tbody>
</table>

Tweet (+)

Tweet (+)

Tweet (+)

Tweet (+)
Input: POS patterns of a tweet

Output: Simple Adjectives

Identification of simple Adjectives Type:
If endcase.suffix = அ ‘a’
Root ≠ verb
then
Return simple adjective

In this example, the adjective term ‘நல்லா’ (good) is used to describe the positive sentiment in a user tweet.

6.3.2 Example for Derived Adjective

அஜிதின் மோசாமணா படம் # பிலா

(Ajithin mosamaana padam #Billa)
(English: Ajithin worst movie # Billa)

<table>
<thead>
<tr>
<th>Tweet</th>
<th>அஜிதின் மோசாமணா படம் # பிலா</th>
<th>(Ajithinmosamaana padam #Billa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokens</td>
<td>அஜிதின்</td>
<td>மோசாமணா</td>
</tr>
<tr>
<td></td>
<td>(Ajithin)</td>
<td>(mosamaana)</td>
</tr>
<tr>
<td>Parts</td>
<td>Noun</td>
<td>Adjective</td>
</tr>
<tr>
<td>Polarity</td>
<td>Negative</td>
<td></td>
</tr>
</tbody>
</table>
Input: POS patterns of a tweet

Output: Derived Adjectives

Identification of derived Adjectives Type:
If endcase.suffix = ஆன ‘aana’
Root ≠ noun
then
Return derived adjective

In this example, the adjective term ‘ஒன்றையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையையைy (English: worst) is used to describe the negative sentiment in a user tweet. In both the examples, the adjectives are occurred before noun and also carry user sentiment.

Table 6.2 Adjective rules for Tamil Sentiment Analysis

<table>
<thead>
<tr>
<th>Structure</th>
<th>Rule</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Suffix – எ (a) ஆன (aana)</td>
<td>நல்லா (nalla)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>அழகானா (Azhagaana)</td>
</tr>
<tr>
<td>Proper</td>
<td>Person or Thing</td>
<td>இந்தியராவம் (India raanuvam)</td>
</tr>
<tr>
<td>Possessive</td>
<td>Suffix – உள்ள (udaiya)</td>
<td>உள்ள (udaiya)</td>
</tr>
<tr>
<td>Future tense</td>
<td>Suffix – உம் (um)</td>
<td>உம் (Endrum)</td>
</tr>
<tr>
<td>Quantifiers</td>
<td>Preposition before noun</td>
<td>சில (sila), காட்டாணை (Evvalavu)</td>
</tr>
<tr>
<td>Intensifiers</td>
<td>Adjectival phrase – மிக (Miga)</td>
<td>மிக (Very good)</td>
</tr>
<tr>
<td>Modifiers (Comparative Proposition)</td>
<td>comparative elements – விட (vita)</td>
<td>(Kabaliyai vIta appa nalla padam)</td>
</tr>
<tr>
<td>Big type</td>
<td>Pronominalize அது (athu), அவை (avai)</td>
<td>அது (nallathu), அவை (kettavai)</td>
</tr>
</tbody>
</table>
In this research, adjectives are treated as a separate grammatical or lexical category to find the sentiment of user tweets. The reason for choosing adjective is that it is the only category which does not co-occur with any clitics. In modern Tamil, eight rules have been identified for adjectives in the field of sentiment analysis task. The adjective rules are derived based on the several occurrences of adjectives in user tweets. These rules are discussed below in Table 6.2.

The patterns for the different combinations of adjectives are mentioned above and are given with sample tweets. All the tweets are derived from the same movie ‘Kabali’.

Example 1: நல்ல கமர்ஷியல் படம் (nalla commercial padam)
(English: Good commercial movie)

Example 2: மேமாசமான அரசியல் காதை (mosamana arasiyal kadhai)
(English: bad politics story)

Example 3: மலேசிய நாட்டில் மைற்று அரசியல் காதை (Malaysia nattin maraimuga arasiyal kadhai)
(English: Malaysia’s indirect politics)

Example 4: என் ஆடைய கட்டுச்செல் ஹிட் (ennudaiya commercial hit)
(English: My commercial hit)

Example 5: கபாலி வாள் தோட்டம் (kabali vasool thodarum)
(English: Kabali collection continues)

Example 6: ராஜியின் மிக நல்ல சண்டைய படம் (Rajniyin miga nalla sandai padam)
(English: Rajni’s very good action movie)

Example 7: கபாலியை வித்து அப்பா நல்ல குடும்ப படம் (kabaliyai vita appa nalla kudumbam padam)
(English: Appa is a good family movie than Kabali)
The following examples show how the sentiment category is populated from the same movie. The Sentiment value calculation is given below.

**For each tweet**

1. If the adjectives are related to the categories, add temp adjective score to the temp category score.
2. Apply the grammar rules to modify the temp category score.
3. Update the category score.

**Movie name:** Kabali

<table>
<thead>
<tr>
<th>Example Tweet</th>
<th>Adjectives/Verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>அதிரடி படத்தாக வந்திருக்கிறது (Athiradi padamaga vanthirukkirathu)</td>
<td>அதிரடி ‘athiradi’, வந்திருக்கிறது ‘vanthirukkirathu’</td>
</tr>
</tbody>
</table>

**Initial score for adjectives/verbs:** \(+1,+1\)

The opinion words அதிரடி ‘athiradi’, வந்திருக்கிறது ‘vanthirukkirathu’ are extracted from the tweet and assigned a value of +1 using rule 1.

**Grammar modifier:** Temporary Scores are \(+2\)

The temporary scores are derived by applying rules 2 and 3. The initial scores are not changed after the rules have been applied because there are no other opinion words or negation terms in the tweet.

**Final scores:** \(+0.66\)
The final scores are normalized in the range from +1 to -1. For example in this case, the final score is calculated using two divided by three.

\[ \text{Final score} = \frac{\text{Adjective score}}{\text{Total number of words in a sentence}} \]  \hspace{1cm} (6.1)

**Overall Polarity** : Positive after applying rules and normalization.

**Calculation of category match:** சண்ைட் ‘Action (+)’

The adjective term அதிராி ‘athiradi’ is closely related to the category of action genre and this tweet is added to the genre count of action.

**Annotation from the Human Expert:** Action (+)

**Accuracy** : 100%

Here in this example, the category outcomes of both the system generated category and human expert annotation are matched. The accuracy is 100% for this tweet.

Similarly, consider another example from the same movie. This example shows how the categories are populated using negation terms.

**Movie name**: கபாலி ‘Kabali’

**Example Tweet** : கபாலி நல்ல குடும்ப படம் இல்லை (Kabali nalla kudumba padam illai)

(English: Kabali not a good family movie)

**Adjectives/verbs** : நல்ல ‘nalla’, குடும்ப ‘kudumba’, இல்லை ‘illai’

**Initial score for adjectives/verbs**: {+2,-1}

**Grammar modifier** : {-2} the polarity is reversed due to negation

**Final score** : {-0.4}

**Overall Polarity** : Negative

**Calculation of category match** : ‘குடும்ப’ Family (-)

**Annotation from the Human Expert**: ‘குடும்ப’ Family (-)

**Accuracy** : 100%
In the above example, the sentiment words நல்ல (good) and குறிப்பிட்டம் (family) are extracted from the tweet and assigned a value of +2 using Rule 1. Here the first adjective acts as an intensifier to the second adjective. Using Rules 2 to 4 the final score is calculated and the category is also matched.

In both the examples shown above, the classification accuracy is 100%. For an example, consider the following tweet where the category match is different.

**Example Tweet**: கமர்ஷியல் படம் தான் எனக்கு பிக்கல (Commercial padam than enakku pidikkala)

**Adjectives**: கமர்ஷியல் ‘commercial’

**Initial score for adjectives**: {+1}

**Final Score**: {-0.2}

**Overall Polarity**: Negative

**Calculation of category match**: கமர்ஷியல் ‘commercial (-)’

**Annotation from the Human Expert**: கமர்ஷியல் ‘commercial (+)’

**Accuracy**: 0 %

In the above example, the final score of the adjective is negative when using the system generated category. However, the actual sentiment category is positive from human annotation thus the accuracy is mismatched.

### 6.4 Algorithm for Patterns of Adjectives

Figure 6.1 shows the process involved in the patterns of adjective rules for sentiment classification. First, from each tweet, all the adjectives and verbs are identified by the POS tagger and considered as potential words for the sentiment classification in the tweet. The scores of the adjectives are populated using the rules 1 to 4 mentioned earlier. Then, any adjectives not within a definite vicinity of a sentiment category were removed.
In addition to adjective scoring, negation handling is also performed. The words with POS tag “NEG” and its types are also used to negate the polarity of adjectives in the score calculation. Final scores of adjectives are normalized between -1 to +1 to avoid biasing over different tweets of length. This is also desired that while dealing with the huge dataset, these values are of great use in analyzing the statistical measures. The algorithm for adjective score calculation is shown in Figure 6.2.

Tamil tweets have a common pattern that is Adjective-Noun-Verb structure. The adjectives are considered as the parts of speech tag that gives the polarity as well as the genre category. However, the verbs and nouns are also used to indicate the direction of the polarity and the specific context respectively. These properties are made use of as a heuristic.

Figure 6.1 Flow diagram for patterns of Adjectives
Calculate Sentiment values of adjectives

Initialize
Read each tweet from corpus
For Each Tweet {
Use POS tags to get syntactic category
If (Negation is present) {
Identify the type of negation and Polarity using negation rules }
Else
For Each Tweet {
Use grammar rules to identify the Adjective type and
Calculate the sentiment values using the rules for adjectives }
} }
End }

**Figure 6.2 Algorithm for assigning sentiment scores to Adjectives**

In this work, the effort has been put on developing the patterns of adjectives and rules for adjectives that classify most of the Tamil tweets in the dataset. An algorithm is also developed using this patterns to calculate the user sentiments. The most commonly occurred patterns of adjectives in Tamil tweets are given in Table 6.3.

**Table 6.3 Most repeating patterns of adjectives in Tamil tweets**

<table>
<thead>
<tr>
<th>Patterns of Adjectives</th>
<th>Example Tweets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjective -&gt; Noun -&gt; Verb (ANV)</td>
<td>அதிரடி படமா வந்திக்கிறான் (Athiradi padama vanthirukkirathu)</td>
</tr>
<tr>
<td>Verb -&gt; Noun -&gt; Adjective (VNA)</td>
<td>பார்க்கப்பட்ட அறை (parka padam arumai)</td>
</tr>
<tr>
<td>Noun -&gt; Verb -&gt; Adjective (NVA)</td>
<td>திரையில் பார்த்தேதன் மகிழ்ச்சி (Thiraiyil parthen magilchi)</td>
</tr>
<tr>
<td>Verb -&gt; Adjective -&gt; Noun (VAN)</td>
<td>பார்த்தரசித்த படம் (Parthu rasitha padam)</td>
</tr>
<tr>
<td>Adjective -&gt; Verb -&gt; Noun (AVN)</td>
<td>மகிழ்ச்சியாகேபாகிறின் படம் (Magilchiyaga pogirathu padam)</td>
</tr>
<tr>
<td>Noun -&gt; Adjective -&gt; Verb (NAV)</td>
<td>படத்தைதரசித்தபார்த்தல் (padathai rasithu parungal)</td>
</tr>
<tr>
<td>Noun -&gt; Adjective -&gt; Noun (NAN)</td>
<td>அஜிதன் நல்லபடம் (Ajithin nalla padam)</td>
</tr>
</tbody>
</table>
Syntax parser identifies and calculates the polarity of a tweet using negation extraction described in the previous Chapter. In order to improve the sentiment analysis, the overall grammar rule approach using adjectives is introduced. Sentiment categorizer model uses the POS tagger to identify the patterns of adjectives using the algorithm shown in Figure 6.3.

```
Calculate ADJ_Category Returns Category {
POS tag (Tweet);
ParseTree (Tweet);
    If ParseTree (Tweet) = ANV
        { Polarity (Tweet) = FindPolarity (V, Tweet); }
    Else-if ParseTree(Tweet) = VNA
        { Polarity (Tweet) = FindPolarity (A, Tweet);} 
    Else-if parse tree (Tweet) = [NVA or VAN] 
        { Polarity (Tweet) = FindPolarity (V+A), Tweet; }
    Else-if parse tree (Tweet) = [NAV or AVN] 
        { Polarity (Tweet) = FindPolarity (A+V, Tweet); }
/* to find the category */
    If polarity (Tweet) = [positive or neutral]
        { Category (Movie) += Category(Adjective) }
    Else
        { Category (Movie)-= Category(Adjective)
    }
```

**Figure 6.3 Algorithm for SA using patterns of Adjectives**

The algorithm is developed based on the hypothesis that all the sentiment related tweets are mostly written by adjectives, nouns and verbs. The adjectives itself has sentiment categories. However, noun is used to identify the context of the domain and verb is used to find the direction of the polarity.

An example for adjective patterns is explained with the following example to show how the polarity and category are populated.
Once the pattern of the adjective is identified the verb is used to find the direction of the polarity. In this example, the verbal term 'இற்கிறாக' is giving the direction of polarity for the adjective term 'அதிராயாக'.

<table>
<thead>
<tr>
<th>Tweet</th>
<th>Padam athiradiyaga irukkirathu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokens</td>
<td>padam  அதிராயாக இற்கிறாக</td>
</tr>
<tr>
<td>Patterns</td>
<td>Noun   Adjective     Verb</td>
</tr>
<tr>
<td>Polarity</td>
<td>Positive</td>
</tr>
</tbody>
</table>

In the above example, the verb gives the positive direction to the adjective term and the tweet is added to the genre class of action category.

However, there is another algorithm to identify the sentiments if the tweet doesn’t fall under any patterns of adjectives described above. If the user tweets are having complex sentences using conjunctions or any other complex phrases, the sentiment model will split the sentences using the parser. Finally, the co-occurring words of adjectives are considered to form the patterns of adjectives. Figure 6.4
shows the algorithm used to identify the pattern of adjective to calculate the sentiment.

Identify ADJ_Patterns Returns Category {
Read each tweet from corpus
If (Conjunction terms are present) {
Split the sentences }
For Each Sentence {
Check the patterns of Adjectives are present or not
If (Pattern of Adjective is present) {
Category (Tweet) = Find Category(ADJ_Category);}
Else If (find Polarity of each Sentence)
(Category (Sentence 1) == Negative(Sentence 2) {
Category (Tweet) = NULL }
Else
Category (Tweet) = Category(Sentence 1)+Category(Sentence2);}
Return Category }

**Figure 6.4 Algorithm for identifying the pattern of Adjectives**

**Use case:**

The below example shows how to derive the sentiment category of a tweet using adjective patterns.

वीरम் ஒரு கமர்ஷியல் படம் ஆனால் நல்ல படம் இல்லை.

(English: Veeram is a commercial movie but not a good movie)

<table>
<thead>
<tr>
<th>விரைவு</th>
<th>ரோம</th>
<th>பொழுதுப்புறம்</th>
<th>படம்</th>
<th>வணக்கம்</th>
<th>படம்</th>
<th>சிலைல்</th>
</tr>
</thead>
</table>

**Figure 6.5 Example for deriving the adjective pattern**

In this example, the sentences will be divided into two using the conjunction term ‘ஆனால் (but)’. After parsing, the adjectives are identified and the word before
and after the adjective term is used to form the adjective patterns. Now using this pattern (NAN) first sentence classified into commercial class but the second sentence has negative term illai which makes the overall polarity as negative. Thus the category of the tweet is assigned as commercial according to the algorithm.

6.5 EXPERIMENTAL RESULTS

Sentiment analysis tool has been developed incorporating all the algorithms based on NLTK and python programming. The tool will automatically display the sentiment values of Tamil movie tweets both at the polarity level as well the genre class. Figure 6.5 shows the sentiment scores for the movie Veeram.

![Figure 6.6 Sentiment analysis results using grammar rules](image)

The performance comparison of grammar rule-based semantic model is given in Table 6.4.
Table 6.4 Performance of Tamil SA using grammar rule

<table>
<thead>
<tr>
<th>Movie</th>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>வீரம் (Veeram)</td>
<td>TF-IDF ranking</td>
<td>27.13</td>
</tr>
<tr>
<td></td>
<td>TF-IDF + DST</td>
<td>36.42</td>
</tr>
<tr>
<td></td>
<td>Tweet Weight</td>
<td>40.26</td>
</tr>
<tr>
<td></td>
<td>Negation rule</td>
<td>47.32</td>
</tr>
<tr>
<td></td>
<td>Adjective rule</td>
<td>60.84</td>
</tr>
</tbody>
</table>

The results suggest that the overall grammar rule based sentiment model produce better performance than other syntactical models. The results also show that accuracy of the sentiment analysis task increases significantly when incorporating the semantic models as features in addition to standard features such as unigrams (TF-IDF). When comparing the performance of grammar rules with other sentiment models, proposed grammar based approach understand the semantic structure of the user sentence in a particular domain.

Figure 6.7 Performance comparison of grammar based semantic models with TF-IDF
Semantic model based on Grammar rules produce a better average accuracy of 64.72% than TF-IDF and other syntactical models as shown in Figure 6.6. Sentiment model has analyzed the tweets and find the polarity as well as the genre categories using proposed grammar rules as well as other algorithms.

The results indicate that the overall grammar considering negation rules and adjective rules is better as it considers complex sentence and incorporates the semantic structures better. The proposed grammar rules handle any type of sentence (simple, compound and complex) in tweets to find the sentiments.

6.6 CONCLUSION

In this section, a semantic model based on an adjective rule is applied on domain dependent tweets using proposed algorithms. The main improvement of the sentiment model is that it understands the domain context of a keyword in natural language is an unsupervised model and therefore does not require any training dataset. The grammar based sentiment model outperforms the baseline model and other syntactic models. However, there are still a lot of challenges to be resolved in the field of sentiment analysis in natural languages.