Domain based Sentiment Analysis in Regional Language-Kannada

Rohini V
(Mtech)
Dept.of CSE, Don Bosco Institute of Technology,VTU, Bengaluru-74.

Merin Thomas
Asst.Prof, Dept .of CSE, Don Bosco Institute of Technology,VTU, Bengaluru-74.

Dr. Latha .C. A
Professor, Dept.of CSE, Global Academy of Technology,VTU, Bengaluru-98.

***Abstract*— Sentiment analysis (SA) is a research field under natural language processing which deals with the study of analyzing opinions or sentiments in the text. Sentiment analysis has gained its momentum by the increase of social networking sites and availability of huge online data. Data mining helps in extracting knowledge from the huge data. Sentiment analysis is incessant in prominent Languages like English .There is very few research work done in regional languages. This Paper focuses on domain based sentiment analysis in Regional language specific movies and also provide a comparison between analysis using direct Kannada dataset and machine translation in English.**

Keywords—*feature extraction, opinion mining, term frequency formatting, decision trees.*

I. INTRODUCTION

In the current system, research works done in sentiment analysis for regional language such as Hindi, Bengali etc are very few compared to English language. These languages were analyzed by comparing lexicon words using part of speech tagging method. This represents emotions as adjectives pattern suffixed word. These semantic words are rated through machine learning approach. Machine learning includes algorithms such as rule based algorithm, maximum entropy etc. India being multi-linguistic nation sentiment analysis in English forms a drawback. For analyzing the documents or websites that are of particular regional language such as Kannada language, it required to have sentiment analysis by using datasets in Kannada. Websites like newskannada, prajavani, wedunia.com, and oneindianews have reviews in Kannada which has a higher preference than the common languages. Sentiment analysis in Kannada with English translation produces similar results with generic set of sentiment words but does not consider the domain of the text which is the feature of an entity called subjective in sentiment analysis. This approach of domain based sentiment analysis is proposed for Kannada language. The work aims in detecting opinion words respective to a particular domain of interest. This research concentrates in domain movie. The result is explored using decision tree classifier technique along with the sentiment scores of the opinion words from lexicon based approach.

II. SENTIMENT ANALYSIS

Sentiment analysis is the task of identifying the orientation of opinion words in a text. Sentiment analysis can be of three levels: document level (such as blog), sentence level (such as comments) and word level. This paper deals with document level Sentiment analysis.

Sentiment analysis can be classified into two approach lexicon based and machine learning. Lexicon based approach deals with comparing sentiment words with the seed words. Lexicon based has two branches Corpus and Dictionary based approach. Machine Learning is a field wherein the system learns from prior sample examples. Here hybrid methodology is implemented incorporating both lexicon and machine learning techniques.

III. LITERATURE SURVEY

Sentiment Analysis based on lexicon approach provide sentiment score in the form of polarities. On occurrence of negation words in a text, simple Lexicon based approach reverses their polarities [1]. Sentiment analysis using Machine Learning approach produces results by either supervised or unsupervised methods. Supervised methods analyze the text by algorithm such as naïve-bayes, maximum entropy [2]. Unsupervised methods follow the point wise mutual information and patterns given by Turney's semantic Orientation method [3]. Lexicon based methods with sentiment polarity produce accurate results compared to bag of words approach. Features are the key attributes in Sentiment analysis, specifying opinions of a user on a particular feature [4]. Sentiment analysis in Hindi language is done using semantic method with a list of Hindi sentiment words and its polarities [5]. Survey of sentiment analysis algorithms in Kannada Language, it compares the algorithms using generic set of words, [6]. Domain based sentiment analysis is achieved by detecting aspects along with opinions for the respective aspects. [7]

A. SYSTEM ARCHITECTURE

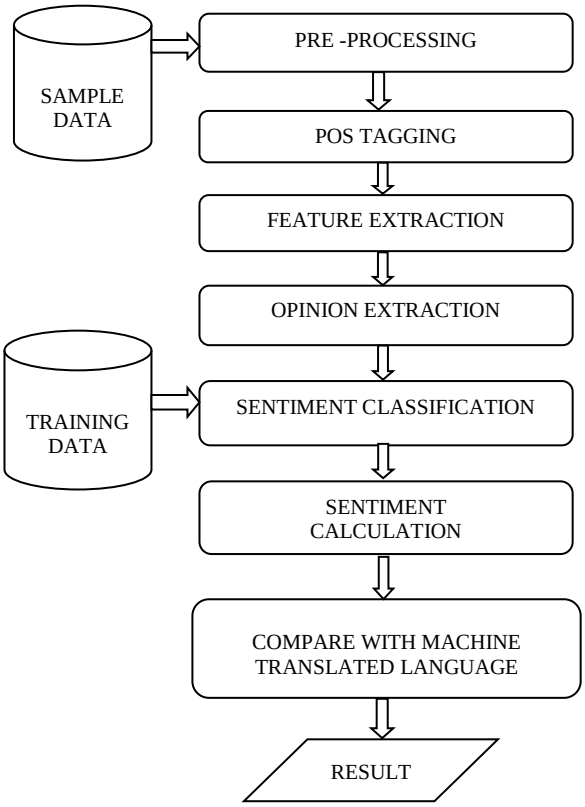

Figure 1: Architecture of Sentiment Analysis Using Decision Trees.

Sentiment analysis follows the steps as shown above. Sample data is the test data that is to be analyzed by the sentiment classifier. Classification requires previously learned labeled data in a particular form Known as Trained data, which are attributes and its values along with target function called Class Label. Text is tokenized during pre-processing of test data and attaches tags to each tokenized words using part of speech taggers. These tagged words are identified as features or opinion words according to role in their text. Sentiment classifier (Decision tree classifier) with the training data classifies the test data into different classes as labeled in the test data and terminates with a rated review as the result of analysis.

B. Dataset Collection

Domain based Sentiment analysis is currently focused on reviews from websites dedicated to movies. Training dataset is framed by a collection of around 100 movie reviews from Kannada websites such as www.vijayakarnataka.com; www.prajvani.com etc. In Sentiment analysis reviews are labeled to classify them into classes

IV. FEATURE AND OPINION WORDS EXTRACTION

Features are the pivot of feature based. It also pivotal domain based sentiment analysis in elucidating the entity to which it belongs to. Features are integrant of an entity (domain). Features also known as aspect define the subject of a text. For example consider:

"ನಿರ್ದೇಶಕ ತುಂಬ ದೊಡ್ಡ ಪ್ರಯತ್ನ ಮಾಡಿದ್ದಾರೆ"

Here ನಿರ್ದೇಶಕ is subject of text

Input data is configured to feature, value pairs in the training data set and appended by the class label for the values. These are also known as attribute-value pairs in sentiment analysis.

Feature extraction is anteceded by pre-processing and tagging.

A. Pre-Processing

Pre-processing implicates stop words removal. Words that do not deliver any meaning for the text are removed known as Tokenization. A list of about 50 stop words was included for tokenization. For example stop words in Kannada language are "ಇದು"," ಎಂಬ"," ಅವನು" etc

B. Part of speech tagging

Each tokenized word is appended with a label known as tag by Part of Speech Tagger. The different type tags are namely: NN, JJ, RB etc for noun, adjective, adverb respectively. Kannada Text is tagged by the Kannada pos tagger by Siva Reddy [9]. For example given a text such as: "movie is good" in Kannada, on tagging it gives the following,

ಚಲನಚಿತ್ರ_NN ಉತ್ತಮ_JJ

C. Feature Extraction

Subjective words in a text are tagged as noun by pos tagger. These are the words that form key points for feature extraction. In feature extraction subjects may be stated directly or indirectly in a text known as Explicit or Implicit feature extraction. *Explicit Feature Extraction: The task of extracting subjective words directly available in text is known as Explicit feature extraction. For eg:*

"ನಿರ್ದೇಶಕ ತುಂಬ ದೊಡ್ಡ ಪ್ರಯತ್ನ ಮಾಡಿದ್ದಾರೆ"

Here the word ನಿರ್ದೇಶಕ feature of text

Implicit Feature Extraction: The task of extracting subjective words mentioned indirectly in a text is known as Implicit feature extraction. For example

ಬೇರೆ ನಿರೂಪಣೆ ಶೈಲಿಯನ್ನು ಹೊಂದಿದ್ದರು "

Though the subject is about story of the movie, but the text mentions about the narration of the movie.

Term Frequency (***tf***): Feature extraction is based on the term count or frequency number known as term frequency. Term Frequency classifies text (documents) into relevant and irrelevant terms.

TF (t) = (Number of times term t appears in a document) / (Total number of terms in the document).

Term Frequency is represented as *tf (t, d)*, i.e. the number of times the term t occurs in a document d. Then raw frequency of t is denoted by $f_{t,d}$ then term frequency is

$$\mathbf{tf\ (t, d)} = f_{t,d} \quad \text{- (1)}$$

where *tf* is the training frequency

t is term whose frequency is to calculated

d is the document

$f_{t,d}$ ***is the*** *raw frequency*

Inverse Document Frequency (IDF): This measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

IDF(t) = log(Total number of documents / Number of documents with term t in it).

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad \text{- (2)}$$

where *t* is term whose frequency is to calculated

d is the document that contains the term t

D is the total number of document that contains the term t

N is the total number of documents in the dataset

Feature Selection: Feature extraction of a document is followed by the process of selecting features with threshold value greater than 0.1. Term frequencies with more than 10 documents are considered as features of the domain. Selected features are grouped into relevant features of the domain.

D. Opinion Word Extraction

Opinion mining or Sentiment analysis identifies and extracts opinion words in a text. These sentiment words relate the opinion of the viewer toward an entity of a movie domain. Similarly these describe opinion of users on a product or attitude of a speaker in politics. Opinion words are identified by part of speech tags, wherein these are suffixed or prefixed by JJ tag relating it as an adjective of the text. [10]

Polarity of opinion words are used to analysis the sentiments of the text. Kannada sentiment words were manually collected and assigned polarities to each word. Certain sentiment words and its polarities are shown below:

ಅದ್ಭುತ	3
ಉತ್ತಮ	4
ಕೆಟ್ಟ	-2
ಶ್ರೇಷ್ಠ	5
ಅಗ್ಗವಾದ	-4

Table 1: Polarity of Kannada sentiment words

V. MACHINE TRANSLATION TO ENGLISH

Machine translation of Kannada Language to the common language English is performed using Google translator tool.

A. Feature Extraction

Pre-processing is initialized by Tokenization. Words that do not deliver any meaning for the text such as ‘a’, ’the’, ’is’ known as stop words are removed. Each tokenized word is appended with a tag by Monty Part of speech tagger from natural language processing kit. The different types of tags are NN, JJ, RB etc, for noun, adjective, adverb respectively. For example, given a text such as: “movie is good”, on tagging, it gives: “movie_NN good_JJ”.

Features are subjective words in a text, tagged as noun by pos tagger. These words form the attribute of an entity and thus the feature to be extracted. In feature extraction there are two types of extraction, explicit and implicit feature extraction. Explicit Feature Extraction: directly available attributes. e.g.: *“mobile price is too high”*. Implicit Feature Extraction: text that indirectly mentioned attribute. e.g.: “*mobile needs to be charged frequently”*

Term Frequency represented as *tf (t, d)*, is the number of times the term t occurs in a document d denoted by $f_{t,d}$. By computing Inverse Document Frequency, frequent terms are weighed down while rare ones are scaled up. Features are selected with threshold value greater than 0.1. Selected features are grouped, to form relevant features of the domain. Opinion words are identified by Part of Speech Tagging. Polarity of opinion words are used to analysis the sentiments of the text. Sentiment words and their polarities were collected from sentistrength.

The training data set consist of features, their values and the class label (positive or negative), here features are represented as F1, F2…Fn under respective documents as shown below:

Doc	F1	F2	F3	F4	F5	F6	Class
1	-2	5	2	1	-3	-3	pos
2	3	-1	-2	3	4	4	pos
3	5	1	1	-2	-4	-4	pos
4	4	-2	-4	3	-1	-1	neg
5	-3	2	-3	-1	-3	-3	neg
6	-1	-3	-1	5	1	1	pos
7	1	-4	2	-4	3	3	pos
8	5	4	3	-2	5	2	neg
9	3	4	5	1	3	5	pos
10	-2	2	2	4	3	1	neg
11	4	5	-1	-3	4	1	neg
12	5	1	4	-2	5	-2	pos
13	3	3	-2	3	1	2	pos
14	3	1	4	2	3	-3	pos

Table 2: Training data set

C. Decision Tree Classifier

Decision trees classifier is powerful tool for classification It forms a tree like structure and represent rules, which can be understood by humans. Decision trees classify instances from the root of the tree, until a leaf node is reached. A decision tree follows divide and conquer method wherein it recursively partitions the instance space. Decision trees were invented by Ross Quinlan.

In a decision tree, each internal node splits the instance space into two or more sub-spaces according to a certain discrete function of the input attributes values. The instance space is partitioned according to the attribute’s value. In the case of numeric attributes, the condition refers to a range. Each leaf is assigned to one class representing the most appropriate target value. Each node is labeled with the attribute it tests, and its branches are labeled with its corresponding values.

In decision tree learning, ID3 alogrithm is used to generate a decision tree from a training dataset.The Decision tree is as shown below where each nodes are features of the entity and leaf nodes are of class pos/neg are represented as Y/N in this graph.

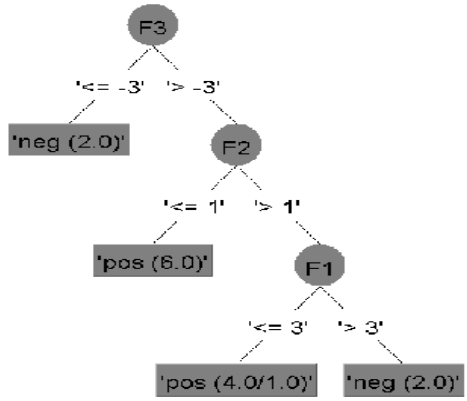

Figure 2: Decision tree Hierarchy (F1,F2…Fn are features).

Algorithm: Decision tree-Classifier algorithm for Kannada

ID3 (Examples, Target_attribute, Attributes)

Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree.

1. Create a Root node for the tree
2. If Examples are positive, Return single-node tree, with label=+
3. If Examples are negative, Return single –node tree, with label=-
4. If Attribute is empty, return the single-node tree Root, with label = most Target_attribute in Examples
5. Otherwise Begin
   6. A ← the attribute from Attributes that classifies Examples
   7. The decision attribute for Root ← A
   8. For each possible value v_i, of A
      - Add a new branch below Root, corresponding to test A= v_i
      - Let Examples v_i be the subset that have value v_i for A
      - If Examples v_i is empty
        - Then below this branch add node with label = most common value of Target_attribute in Examples
      - Else below this new branch add the subtree
        - ID3 (Examples v_i, Target_attribute, Attributes-[A]})

   End
   Return Root

The above algorithm is the ID3(Iterative Dichotomiser 3) algorithm used for buiding decision tree in sentiemtn analysis and this algorithm is developed in Python for the sentiemtent analysis

Decision tree induction is closely related to rule induction. Each path from the root of a decision tree to one of its leaves can be transformed into a rule simply by conjoining the tests along the path to form the antecedent part, and taking the leaf's class prediction as the class value. The resulting rule set can then be simplified to improve its comprehensibility to a human user. Each internal node denotes a test on an feature, each branch denotes the outcome of a test, and each leaf node holds a class label.[8]

Feature with highest information gain is considered as splitting node or root node for the formation of decision tree and feature values form the branches of the node. Each time gain is calculated to build decision tree for training data set.Splitting node or the root node is calculated though information gain as shown below for each feature in data training set.

$$Gain\ (S, A) = Entropy\ (S) - Entropy\ (S, A) \quad - (3)$$

The target attribute can take on different values, Let p_i be the proportion of S belonging to class $_i$ then entropy of S is defined as:

$$Entropy(S) = \sum_{i=1} - p_i\ (I)\ log2\ p_i\ (I) \quad - (4)$$

$$Entropy(S, A) = \sum ((|S_v| / |S|) * Entropy\ (S_v)) \quad - (5)$$

where values (A) is set of all possible values for attribute A
S_v is the subset of S for which attribute A has values.

D. Comparison

Sentiment analysis of kannada text and machine translated english language produces the following output as shown below:

Language	*Input Text*	*positive*	*negative*
Kannada	ಈ ಚಲನಚಿತ್ರ ತುಂಬಾ ಚೆನ್ನಾಗಿತ್ತು	Yes	-
English	This was a very good movie	Yes	-
kannada	ಈ ಚಿತ್ರದು ಕಥೆ ಚೆನ್ನಾಗಿಲ್ಲ	-	Yes
English	The story of this film is not good	-	Yes

Table 3: Comparison Kannada and English text sentiment analysis

	Test Data	*Precision*	*Recall*
1	Kannada test data	0.78	0.79
2	English test data	0.86	0.67

Table 4: Results

CONCLUSION

Sentiment analysis using Semantic method with opinion Lexicons helps in extracting subjective word called aspect which defines attribute of the particular entity (domain) of the text. There are many approaches to classify the features and opinion in English language compared to Indian regional languages. Features based opinion mining is a challenging task in Kannada language and there are few tools available such as Training data set specific to a domain, part of speech Tagger.

Certain words in Kannada like "ಕಥೆಗೊಂದು", "ಎನ್ನುವುದಕ್ಕೂ", "ಮನೆಯೊಂದರಲ್ಲೇ","ಚರ್ಚೆಯಾಗುತ್ತವೆ"on machine translation produced ambiguous text, which forms one of the major causes of improper results. Thus analyzing test data in regional language gives better accurate results compared to Machine Translated English Language.

REFERENCES

[1] Prabhu Palanisamy, Vineet Yadav and Harsha Elchuri, "Serendio:Simple and Practical Lexicon Based approach to Sentiment analysis," Serendio Software Pvt Lid, Guindy, Chennai
[2] I.Hemalatha, Dr.G.P Saradhi Varma, Dr.A.Govardhan,"Sentiement analysis tool using machine learning alogrithm",volume 2, Issue 2, International journal of emerging Trends and Technology in computer science
[3] Peter D.Turney,"Thumbs Up and Thumbs Down? semantic orientation applied to Unsupervised Classification of Reviews",Institute for Information Technology, National Research Council of Canada Ottawa,Proceeding of $oth Annual Meeting of the Association for computional Linguistics Philadelphia,July 2002.
[4] Solamki Yogesh Ganeshbhai,Bhumika K. Shah,"Feature based opinion mining:A Survey",Surat ,India,2015, IEEE.
[5] Aditya Joshi, Balamurali A R, Pushpak Bhattacharyya,"A Fall-back Strategy for Sentiment Analysis in hindi: a Case Study",IIT Bombay.Proceedings of ICON 2010: 8th International Conference on Natural Langugae Processing
[6] K.M.Anil Kumar, N.Rajasimha "Analysis of user's sentiment for kannada web documents",Sciencedirect, Procedia computer science 54 (2015) 247-256, Eleventh International Multi-Conference on Information Processing.
[7] Ankit singh, Md.Enayat Ullah,"Aspect based Seniment Analysis",Indian Institute, Kanpur,CS365A,http://goo.gl/qdTRhf.
[8] Jeevanandam Jotheswaran, Dr.Y.S.Kumaraswamy,"Opinion mining using decision tree based feature selection through manhattan hierarchical cluster measure" Journal of theoretical and applied science,volume 58,issue no 1,2013.
[9] http://sivareddy.in/downloads
[10] Tien-thanh Vu,Huyen-Trang Pham, Cong-To Luu,"A Feature-Based Opinion Mining Model on Product Reviews in Vietnamese",Vietname National University, Hanoi(VNU), College of technology,Semantic Methods,SCI381, pp.23-33,springerlink.com, 2011