A Peer Reviewed
TECHNICAL JOURNAL Vol 2, No.1, October 2020
Nepal Engineers' Association, Gandaki Province
ISSN : 2676-1416 (Print)
Pp.: 22-29

ASPECT BASED SENTIMENT ANALYSIS OF NEPALI TEXT USING SUPPORT VECTOR MACHINE AND NAIVE BAYES

Sujan Tamrakar[1,*], Bal Krishna Bal[2] and Rajendra Bahadur Thapa[3]
[1,3]Gandaki College of Engineering and Science, Pokhara University, Nepal
[2]Department of Computer Science and Engineering, Kathmandu University, Nepal
**E-mail: sujan@gces.edu.np*

Abstract

Aspect-based Sentiment Analysis assists in understanding the opinion of the associated entities helping for a better quality of a service or a product. A model is developed to detect the aspect-based sentiment in Nepali text using Machine Learning (ML) classifier algorithms namely Support Vector Machine (SVM) and Naïve Bayes (NB). The system collects Nepali text data from various websites and Part of Speech (POS) tagging is applied to extract the desired features of aspect and sentiment. Manual labeling is done for each sentence to identify the sentiment of the sentence. Term Frequency – Inverse Document Frequency (TF-IDF) is applied to compute the importance of the words. The feature vectors thus produced are then applied to the Classifier algorithms to predict and classify the sentence. The accuracy obtained by the SVM classifier is 76.8% whereas Bernoulli NB is 77.5%.

Keywords

Aspect-based; Classification; Machine Learning; Nepali text; Sentiment Analysis; Technology

1. Introduction

With the rise of online platforms to shop, trade, communicate, entertain, educate, an enormous amount of activities are going on in the digital world. Massive amounts of user interactions are available in terms of ratings, sharing, comments, and feedback on those platforms. (Forbes, 2018) indicates 2.5 quintillion bytes of data are generated each day globally. This astonishing amount of data is not only of the most spoken languages like English, Mandarin Chinese, and Hindi but lately of various non-English languages, including Spanish, French, and Nepali. (Nepal, 2019) shows that the total number of active internet users in Nepal stands to be 16.19 million measuring 54% of Nepal's population. This has become possible due to the availability and affordability of internet access, the Introduction of mobile phones and handheld devices, and improvement in hardware devices and software systems. The presence of large data carries lots of information in them. Any stakeholder needs to understand the sentiment and meaning opinionated in the fine-grained user reviews.

Understanding emotion-filled reviews help in understanding the user thereby assisting in satisfying users and providing better quality. (Salinca, 2015 verified that sentiment classification proved to perform well when considering user ratings. Nevertheless, to judge user opinion, rating only does not provide reliable information. (Bose, Dey, Roy, & Sarddar, 2018) demonstrated that user ratings and comments can conflict. To understand the user response, it is necessary to tap the feedback and analyze them individually. But, extracting sentiments from user response is a crucial yet challenging factor for any firm or individual. This is largely due to the high volume, possibly hundreds and thousands of responses as well as unstructured - non-grammatical - writing style, and having typos. (Seerat & Azam, 2012) mentions challenges like object identification, feature extraction, grouping synonyms, opinion orientation classification, identifying comparison words, different writing styles, opinions change over time, presence of sarcasm & ironic statements and use of double negatives in extracting opinion from user textual data.

In Nepali digital space, one can find content, reviews and feedback that are in the Nepali language. There are three different ways of analyzing sentiments from texts: Document-level, Sentence-level, and Aspect-level. Researches have been performed on detecting sentiments (Gupta & Bal, 2015) and classifying sentiments (Thapa & Bal, 2016) in the document-level for Nepali texts. Document-level emphasizes on providing a single sentiment of an entire document, Sentence-level provides a sentiment of each sentence and Aspect-level gives sentiments of phrases or entities within the sentence. Aspects, generally found to be a noun, noun groups or proper nouns and adjectives are related to being sentiments.

Consider the following Nepali language examples:

यस ट्याब्लेटमा मल्टिमेडिया हेर्नु रमाइलो अनुभव हो ।, मल्टिमेडिया, रमाइलो, 1 ... (1)

यस ट्याब्लेटमा फ्ल्यासको अभाव पनि एउटा कारण हो ।, फ्ल्यास, अभाव, 0 ... (2)

The above sentences induce meanings as shown in Table 1:

Table 1: Aspect Based Sentiment Analysis result.

Sentence	Aspect	Sentiment	Polarity
(1)	मल्टिमेडिया	रमाइलो	Positive
(2)	फ्ल्यास	अभाव	Negative

Aspect Based Sentiment Analysis (ABSA) could be understood as finding the aspect words from sentences and computing the polarity of every aspect. Aspect denotes the component or attribute of an entity. A sentence can include the presence of either a single aspect or multiple aspects. This study is concentrated on working on a single aspect.

2. Methods

2.1 Architecture

Figure 1 shows the high-level system architecture of the study. Initially, data are collated from multiple websites by using a crawler. Data are pre-processed cleaning all the sentences; removing wild card characters and symbols. Afterward, tokenization is done by using Natural Language ToolKit (NLTK) for the Indic Languages (iNLTK) library (Arora, 2019) . iNLTK is used due to its support for Nepali languages along with other Indic languages like Sanskrit, Hindi, Bengali, Urdu, Marathi, etc. An example from the dataset is given below:

यसले गर्दा बिध्यमान फोरजी सञ्जाल मार्फत नै उपभोक्तालाई फाइभजी सेवा प्रदान गर्न सकिने भएको छ ।

The above sentence is tokenized as following using iNLTK Tokenizer:

[‘यस’, ‘ले’, ‘गर्दा’, ‘बिध्यमान’, ‘फो’, ‘जी’, ‘सञ्जाल’, ‘मार्फत’, ‘नै’, ‘उपभोक्ता’, ‘लाई’, ‘फा’, ‘इभ’, ‘जी’, ‘सेवा’, ‘प्रदान’, ‘गर्न’, ‘सकिने’, ‘भएको’, ‘छ’]

Using NLTK Tokenizer, it would result as:

[‘यसले’, ‘गर्दा’, ‘बिध्यमान’, ‘फोरजी’, ‘सञ्जाल’, ‘मार्फत’, ‘नै’, ‘उपभोक्तालाई’, ‘फाइभजी’, ‘सेवा’, ‘प्रदान’, ‘गर्न’, ‘सकिने’, ‘भएको’, ‘छ’]

A set of 52 stop-words are applied to remove the stop-words from the dataset. Thus, generated sentences are then applied to Trigrams'n'Tags (TnT) POS tagger using an enhanced tagged corpus to tokenize and categorize every word to respective POS tags. The result obtained is as following:

यस: DUM	ले: PLE	गर्दा: VBO
बिध्यमान: JJ	फो: Unk	जी: Unk
सञ्जाल: NN	मार्फत: POP	नै: RP
उपभोक्ता: NN	लाई: PLAI	फा: Unk
इभ: Unk	जी: Unk	सेवा: NN
प्रदान: NN	गर्न: VBI	सकिने: VBNE
भएको: VBKO	छ: VBX	

Some of the above POS tags (Prajwal Rupakheti, 2013)

indicate:

DUM = Pronoun (Unmarked Demonstrative)

PLE = Postposition (Le-Postposition)

VBO = Verb (Participle)

JJ = Adjective (Normal)

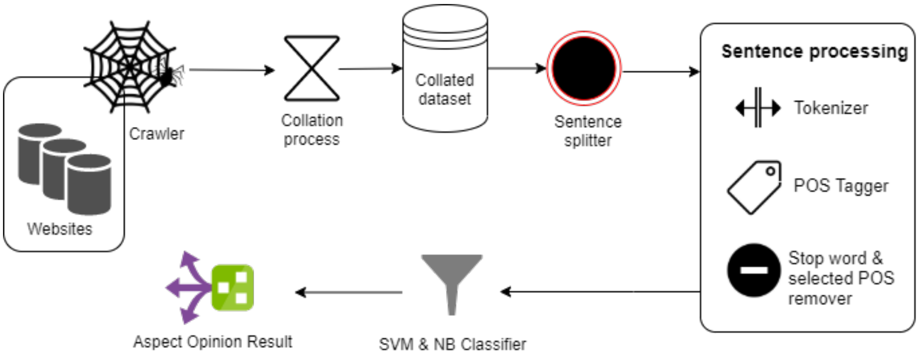


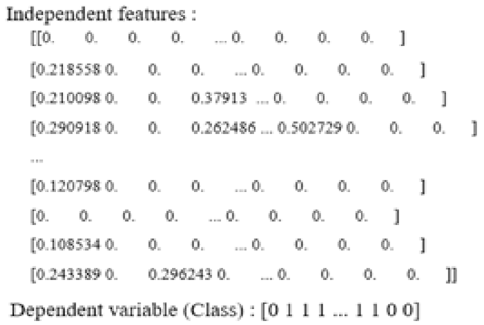
Figure 1: High-level system architecture

To narrow down the scope and emphasize on only limited POS, this study has undertaken two hypotheses, viz:

1. Aspect is indicated by noun words.
2. The sentiment of the aspect is represented by adjective words.

Training data has to be transformed into something a machine can understand i.e. vectors. Therefore, the tokenized word is converted into numerical vectors measuring the weightage of words using TF-IDFVectorizer which turns a collection of text sentences into a numerical feature vector.

Figure 2 shows the generated vector after applying TF-IDFVectorizer.



```
Independent features :
  [[0.        0.        0.        0.        ... 0.        0.        0.        0.       ]
   [0.218558 0.        0.        0.        ... 0.        0.        0.        0.       ]
   [0.210098 0.        0.        0.37913  ... 0.        0.        0.        0.       ]
   [0.290918 0.        0.        0.262486 ... 0.502729 0.        0.        0.       ]
   ...
   [0.120798 0.        0.        0.        ... 0.        0.        0.        0.       ]
   [0.        0.        0.        0.        ... 0.        0.        0.        0.       ]
   [0.108534 0.        0.        0.        ... 0.        0.        0.        0.       ]
   [0.243389 0.        0.296243 0.        ... 0.        0.        0.        0.       ]]
 Dependent variable (Class) : [0 1 1 1 ... 1 1 0 0]
```

Figure 2: Result after applying TF-IDF Vectorizer

Independent features represent the feature variables i.e. the weight of the words in the dataset whereas Dependent variables represent the polarity label of the class. Vectorized feature using TF-IDF is given to the classifier algorithms as input. By using vectors, the system can extract relevant features which help it learn from the existing data and make predictions about the texts to come. Afterward, SVM and NB classifier's different evaluation metric scores are recorded obtained using various hyper-parameters.

2.2 Data Collection

Opinions and feedback are scrapped from Nepali websites like (Patro, 2017), (Frame, 2018), (TechPana Media Pvt. Ltd., 2019) and (SailungOnline - Digital Newspaper, 2016) and are used to create a collated dataset. Among numerous categories from the website, only the Technology sector is chosen for this study. A total of 1576 sentences are collected out of which 788 sentences are of positive sentiment whereas 788 are of negative sentiment. Experiments are conducted with a balanced positive-negative sentence dataset. Table 2 shows a sample dataset.

Table 2: A sample dataset containing positive and negative sentences

Sentence	Aspect	Sentiment	Polarity
ट्याब्लेटमा स्पिकर ग्रिलले यसको सुन्दरतामा थप पार्छ।	स्पिकर ग्रिल	सुन्दरता	1
जडानको बारेमा कुरा गर्दै यो एक उत्तम उपकरण हो।	जडान	उत्तम	1
यसको ब्याट्री कमजोर छ।	ब्याट्री	कमजोर	0
यद्यिप हामीले पत्ता लगायौं कि यसको टच कहिलेकाँही नराम्रो छ।	टच	नराम्रो	0

2.3 Algorithms

2.3.1 Support Vector Machine

SVM is a supervised ML algorithm which is used mostly for classification problems. A hyperplane differentiates the data in two classes for binary classification problems as shown in figure 3. In Multiclass SVM, multiple hyperplanes differentiate the data in multiple classes as shown in figure 4.

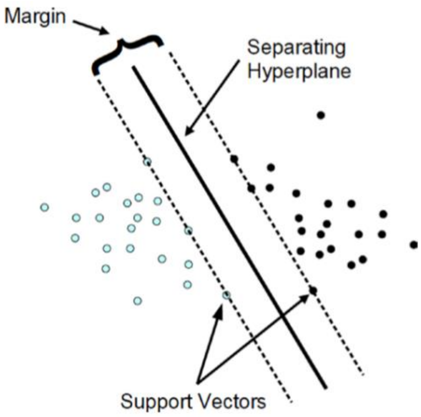


Figure 3: Binary SVM Classifier

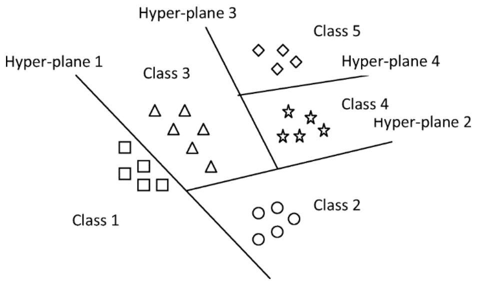


Figure 4: Multiclass SVM Classifier

Nouns, noun groups, or proper nouns are considered as aspects and adjectives to be the related sentiments. These are identified by POS tagging and noun phrases are regarded as the features of the sentence.

Therefore, we have

$$A \to S \qquad \ldots (1)$$

where,

A = aspect of product/service

S = sentiment of that aspect

In this study, sentiments are regarded to be positive and negative and not dealing with neutral sentiment words. Hence, SVM is applied as a binary classifier. A → S is an aspect-sentiment pair wherein 'S' takes polarity values as either positive or negative. The classifier can be trained on a training dataset by using feature set {(x1, y1),…,(xi, yi), ..., (xN, yN)} SVM classifier has to be trained to discover a hyperplane that best classifies the data into two classes.

Consider a dataset containing 'N' features, denoted as xi and each xi has a label yi at an ith position which is either +1 or -1 class among a binary classifier.

i.e. xi = aspects; ex: Vector of मल्टिमेडिया

yi = sentiments; ex: Vector of रमाइलो

yielding +1 class for this particular review.

SVM classifier identifies the aspects' sentiments to fall in either of the classes which is given by:

$$y = c\,\{(\boldsymbol{wx} + b) + \frac{1}{2}\,\left||w|\right|^{2}{}_{2}\} \qquad \ldots (2)$$

where,

w = weight vector, b = bias and

c =regularization parameter

Parameters 'w' and 'b' are calculated based on the gradient descent (Medium, 2019)

The SVM equation of ith aspect training value can be expressed as:

$$if\ w_i x_i + b > 0, y_i = +1$$
$$\text{else}\ if\ w_i x_i + b < 0, y_i = -1 \qquad \ldots (3)$$

The linear kernel is used to classify the sentiments by linearly separating the data. It is due to the presence of higher dimensionality of instances and features that makes linear separation effective.

2.3.2 Naïve Bayes

NB is based on Bayes Theorem and uses probability theory to describe the probability of an event to occur depending on prior information. In ML, Naïve Bayes is a classification technique that supposes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It uses probabilistic analysis to extract features from numerical feature vectors or document-term matrix. These features help in the training of the Naive Bayes Classifier. In the classification problem, every word is served to be a sign of the assigned polarity.

New data 'x' → Naive Bayes Classifier → Class 'c'

$$P(c|x) = P(x|c)P(c) \qquad \ldots (4)$$

P(c|x) = P(x|c)P(c)

In above Bayes Theorem,

P(c) = prior probability

P(c | x) = probability of c at given x

P(x | c) = probability of x at given c

Assuming the training set has 4 positive and 4 negative class sentences. From the count vectorizer counter, the total frequency of words ‘डिजाइन’, ‘एकदम’ and ‘राम्रो’ are 7 (4 positives), 5 (3 positives), 4 (3 positives) respectively. Then,

P(x|positive) = P(positive) × P(डिजाइन | positive) × P(एकदम|positive) × P(राम्रो|positive)

$$= 4/8 \times 4/4 \times 3/4 \times 3/4$$

$$= 0.28 \qquad \ldots (5)$$

Sentence ‘डिजाइन एकदम राम्रो’ outputs 0.28 given positive class. Similarly, the sentence ‘डिजाइन एकदम नराम्रो’ outputs 0.047 given negative class. Thus, by argmax function, 0.28 is chosen to be the predicted class i.e., positive class.

The important work to perform before classification is to find the parameters for the features’ probability distributions. Afterward, for classification, the opinion with the largest probability given the data point’s features is chosen:

$$y = \underset{c_i}{argmax}\, P(c_i) \prod_{j=1}^{n} P(x_j|c_i) \qquad \ldots (6)$$

3. Results Analysis

3.1 Experiment Protocol

Table 3 shows the experimental protocol used for this study:

Table 3: Experiment Protocol setup.

Experiment Parameter	Value
Operating System	Windows 10
Processor	Intel ® Core TradeMark i5-8250U CPU @ 1.60 GHz processor
Random Access Memory	12 GB
Programming Language	Python 3.6
iNLTK library	Default
NumPy	1.16.6
Pandas	0.24.2
Scikit-learn	0.20.4
Scipy	1.2.3

ML classification algorithms like SVM and NB have been implemented during the experimentation. To measure the model performance, evaluation metrics like F1 score, precision, recall, and accuracy are computed.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \times \frac{P \times R}{P + R}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad \ldots (7)$$

where,

TP = True Positive TN = True Negative

FP = False Positive FN = False Negative

3.2 Results

Tables 4, 5 and 6 show the results obtained using SVM, Bernoulli Naïve Bayes (BNB) and Gaussian Naïve Bayes (GNB) algorithm respectively.

3.2.1 Support Vector Machine

The entire experiment is performed using Linear

Kernel due to the reason for having two classes i.e. positive and negative. The regularization parameter 'c' has been chosen to 0.1, 0.3, 1 and 3 during the experiment. Table 4 shows the result thus obtained.

Table 4: Results achieved after applying the SVM algorithm.

c	Accuracy	F1	Precision	Recall
0.1	0.699	0.713	0.727	0.699
0.3	**0.768**	**0.769**	**0.769**	**0.769**
1	0.759	0.759	0.759	0.759
3	0.740	0.740	0.740	0.740

3.2.2 Bernoulli Naïve Bayes

For BNB, during the splitting of the dataset into training and testing set, parameter 'random state' (RS) has been chosen as 0 and 42 for the experiment to control the data shuffling. Table 5 shows the result thus obtained.

Table 5: Results achieved after applying the BNB algorithm.

RS	Accuracy	F1	Precision	Recall
0	0.740	0.740	0.741	0.740
42	**0.775**	**0.777**	**0.780**	**0.775**

3.2.3 Gaussian Naïve Bayes

GNB is experimented to observe the performance of the system. The score parameter 'average' has been chosen as macro and weighted for the experiment. Table 6 shows the result thus obtained.

Table 6: Results achieved after applying the GNB algorithm.

Accuracy	F1	Precision	Recall
0.613	0.628	0.643	0.613

BNB performed better than SVM due to the dataset containing shorter sentences. BNB is better than GNB because it works on the principle of binomial distribution whereas GNB works on Gaussian distribution which works best on linear regression.

4. Conclusion

It has been an important factor for any business or individual to capture the feeling ridden in the user's opinions and reviews. This leads to understanding consumer observation on a granular level thus helping in satisfying consumers. The study shows Machine Learning algorithms like SVM and NB to detect the sentiments in user opinions focused on any particular entity. This method implements a limited dataset (1576 sentences) and shows that BNB performs better than the SVM when conducted on various experiments. For further improvements, the system can be tested against a bigger dataset which could produce an increment in the accuracy. Moreover, fine-tuning hyperparameters in the algorithms could generate a better result. Other ML algorithms like Nearest Neighbor, Decision Trees and Random Forest could also be used for classification.

Acknowledgment

We would like to thank Dr. Ashim Khadka for his continuous assistance. We would also take this opportunity to pay special thanks to Dr. Pramod Parajuli and Er. Mahesh Shakya for providing useful critiques during the initial research phase.

References

Arora, G. (2019). *Natural Language Toolkit for Indic Languages - iNLTK latest documentation*. Retrieved March 7, 2019, from Natural Language Toolkit for Indic Languages: https://inltk. readthedocs.io/en/latest/index.html

Bose, R., Dey, R. K., Roy, S., & Sarddar, D.

(2018). Sentiment Analysis on Online Product Reviews. *ICT4SD 2018, 30 31st August 2018*. DOI: ICT4SD 2018

Forbes, F. (2018, May 21). *How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read*. Retrieved May 15, 2019, from Forbes: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read

Frame, I. C. (2018). *Archives*. Retrieved April 25, 2019, from ICTFrame.com: https://np.ictframe.com/category/gadgets/

Gupta, C. P., & Bal, B. K. (2015). Detecting Sentiment in Nepali texts: A bootstrap approach for Sentiment Analysis of texts in the Nepali language. *2015 International Conference on Cognitive Computing and Information Processing (CCIP)*, (pp. 1-4). Retrieved May 20, 2019

Medium. (2019, May 15). *Learning Parameters, Part 1: Gradient Descent*. Retrieved May 13, 2020, from Towards Data Science: https://towardsdatascience.com/learning-parameters-part-1-eb3e8bb9ffbb

Nepal, D. (2019, January 31). *Digital 2019: Nepal*. Retrieved June 6, 2019, from https://datareportal.com/reports/digital-2019-nepal?rq=nepal

Patro, H. (2017). *Articles Technology | Hamro Patro*. Retrieved October 16, 2019, from Hamropatro.com: https://www.hamro patro.com/posts/articles-Technology

Prajwal Rupakheti, D. L. (2013). Report on Nepali Computational Grammar. Retrieved October 15, 2019, from https://www.researchgate.net/publication/237310273_Report_on_Nepali_Computational_Grammar

SailungOnline - Digital Newspaper. (2016). Retrieved May 19, 2019, from SailungOnline: https://www.sailung online.com/category/प्रबिधि/

Salinca, A. (2015). Business Reviews Classification Using Sentiment Analysis. *IEEE.* Timisoara: IEEE. DOI:10.1109/synasc.2015.46

Seerat, B., & Azam, F. (2012, July). Opinion Mining: Issues and Challenges (A survey). *International Journal of Computer Applications, 49*, 42-51. DOI:10.5120/7658-0762

TechPana Media Pvt. Ltd. (2019). *Gadgets Review Archives - TechPana | Digging into Tech TechPana*. (TechPana Media Pvt. Ltd.) Retrieved May 8, 2020, from TechPana: https://www.techpana.com/ category/gadgets/gadgets-review/

Thapa, L. B., & Bal, B. K. (2016). Classifying sentiments in Nepali subjective texts. *2016 7th International conference on information, intelligence, systems & applications (IISA)*, (pp. 1-6). Retrieved May 15, 2019