Proceedings of IOE Graduate Conference, 2017
Volume: 5 ISSN: 2350-8914 (Online), 2350-8906 (Print)

Trend Analysis of Technology News in Nepali Newspapers. A case study: The Kathmandu Post

Slesa Adhikari [a], Arun Kumar Timalsina [b]

[a, b] *Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, TU, Nepal*
Corresponding Email: [a] slesa.adhikari@ioe.edu.np, [b] t.arun@ioe.edu.np

Abstract
Text classification, a case of Natural Language Processing, assigns one or more class to a text document based on its content. In this paper, text classification is applied to study the trend of news, specifically the coverage of technological news, in one the most popular nepali news providers - The Kathmandu Post. Two methods of classification are explored, namely, Naive Bayes and Support Vector Machines. A pre-categorized news dataset is used to train the model and the accuracy of both the estimators are compared based on a portion of the labeled dataset. The best model is used to classify the news articles from The Kathmandu Post and the trend is analysed. Finally, a projection of the trend in the future is estimated.

Keywords
News Classification – Support Vector Machines – Naive Bayes – Trend Analysis

1. Introduction

News articles are written under variety of topics. Nepali news are mostly common for topics like politics and entertainment. Yet concerns in the field of technology is not completely absent. It is certainly interesting to study how the topics of science and technology are being covered with respect to other topics by a common and popular newspaper, and whether it has been increasing lately.

Text classification has been a common issue of interest in the recent years. It is widely used for applications such as sentiment analysis, language identification and data mining. News categorization can be considered a data mining application and is considerably useful when either the news source does not distinguish the desired category or the given categorization is not much reliable.

The most common approach to text classification is supervised machine learning. First, a classification model is created with initialization variables. Then, a training dataset is selected which contains pre-categorized texts and is used for training the model and finding the optimal values for the variables. Finally, the model is used for actual classification of texts.

This paper tries to use two different machine learning techniques for text classification, which are widely popular: Naive Bayes and Support Vector Machine (SVM). First, the model is trained using a dataset provided by BBC. This dataset has been preclassified into multiple categories like science and technology, politics, sports, and entertainment. Then, the news articles from our local newspaper are fed to the trained model. The Kathmandu Post has an extensive archive available online [1]. The articles from this archive are classified to either of the two categories: science and technology or others. The problem is thus a binary classification problem. Once the classification is done, the results are used to perform a brief analysis of the trend of penetration of technology in the Nepali news scene. Finally, a projection of the trend in the future is also made.

1.1 Related Works

Some scholarly articles can be found with research on similar type of work, though not entirely same. An example is Indonesian News Classification using SVM by Lilian and co [2]. With its result having an accuracy rate of 85%, the paper shows that SVM is pretty good at classifying digital news. Another similar research is by Neeru Sharma and co [3] which uses both Neural

Network and SVM to classify News into four categories and use the result to show that SVM is more efficient. Although several news classification research have been done, none have been found for Nepali digital news. One reason may be lack of digital news in structured format. Trend analysis of technology news in Nepali newspaper is however a completely uncharted area. No source can be found on the internet that has done such analysis whether using machine learning or not.

2. Theory and Techniques used

2.1 Feature Extraction

Almost all machine learning problems require numerical features to be the input - text can not directly be fed. The raw dataset is often too large and consists of many redundant features that do not contribute much (in some cases, can even be detrimental [4]) to the process of classification. Thus, a need for extracting the most useful features from the dataset arises. This is done using feature extraction. Consequently, feature extraction is a process that extracts only the most significant characteristics (much like dimensionality reduction) from the raw text, but these extracted features need to represent the data with sufficient accuracy. It takes as input the raw text of variable length and outputs a vector of features of fixed length which can be fed to the learning algorithms. In this research, the bag-of-words representation is used.

Bag of words Bag of words is a method of feature extraction in which just the words that occur in the text are kept, ignoring all other features such as the grammar, word ordering, word positioning. After the texts are classified into bag of words, other characteristics are associated with each word. The most often used is the frequency of the words, i.e how many times the word appears in the document. For an example, consider two texts:

1. "I am Barry Allen and I am the fastest man alive"
2. "I am the dark. I am the night. I am Batman"

The bag of words associated with the given texts is a list as follows:

["I", "am", "Barry", "Allen", "and", "the", "fastest", "man", "alive", "dark", "night", "Batman"]

Counting in the frequency, the feature vector becomes:

1. [2, 2, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0]
2. [3, 3, 0, 0, 0, 2, 0, 0, 0, 1, 1, 1]

This configuration tokenizes the string by extracting words of at least 2 letters, i.e, words like 'a', 'I' are ignored. Also, the english stop-words (non-essential words) are not extracted. In addition, to preserve some positioning information, unigrams as well as bigrams are used in the feature vectors.

2.2 TF-IDF weighting

In a large text corpus, some words appear more frequently than the others, words such as 'a', 'the', 'of', 'is', 'very' etc. Such words, though not completely useless, most often do not provide meaningful information about the text as a whole and at times even overshadow the frequencies of the rarer but more interesting words. TF-IDF (Term-Frequency times Inverse Document-Frequency) is an approach used to solve this discrepancy. Basically, TF-IDF value of a term is the measure of how important the word is in relation to the document. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. [5] TF-IDF is calculated as follows: [6]

$$tf-idf(t,d) = tf(t,d) * idf(t)$$

where, $tf(t,d)$ is the frequency of the term ' t ' in document ' d ' and $idf(t)$ is calculated as,

$$idf(t) = \log \frac{1+n_d}{1+df(d,t)} + 1$$

where n_d is the total number of documents, and $df(d,t)$ is the number of documents that contain term t.

2.3 Naive Bayes'

Naive Bayes' is the family of probabilistic supervised learning algorithms that are based on Bayes' theorem, but with the "naive" assumption that every pair of features are independent of each other. The probability

that a document with feature vectors x_1 through x_n be classified as class y is given by Naive Bayes' theorem as [7]:

$$P(y|x_1,...,x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

$$\hat{y} = arg\max_y P(y)\prod_{i=1}^{n} P(x_i|y)$$

and Maximum A Posteriori (MAP) estimation is used to estimate $P(y)$ and $P(x_i|y)$. The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i|y)$. The one used in the paper is the Multinomial Naive Bayes classifier.

2.3.1 Multinomial Naive Bayes

Multinomial Naive Bayes classifier is one of the most successful classifiers when it comes to text classification - mostly data that can be turned into counts. It assumes that the data is a multinomial distribution. The distribution is parameterized by vectors $\theta_y = (\theta_{y1},...\theta_{yn})$ for each class y , where n is the number of features (in text classification, the size of the vocabulary) and θ_{yi} is the probability $P(x_i|y)$ of feature i appearing in a sample belonging to class y . The probability is given by,

$$\hat{\theta}_{yi} = \frac{N_{yi}+\alpha}{N_y+\alpha n}$$

where $N_{yi} = \sum_{x\varepsilon T} x_i$ is the number of times feature i appears in a sample of class y in the training set T , and $N_y = \sum_{i=1}^{|T|} N_{yi}$ is the total count of all features for class y .

The smoothing priors $\alpha \geq 0$ accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting $\alpha = 1$ is called Laplace smoothing, while $\alpha < 1$ is called Lidstone smoothing. [7]

2.4 Linear Support Vector Machine

Linear Support Vector Machines (SVM) are non-probabilistic binary linear classifiers that use supervised learning strategy. As such, a SVM model classifies new examples discretely into one of the two categories, unlike Naive Bayes Classifiers that give the probabilistic approximation. Linear SVM can classify linearly separable datasets, which are divided by a hyperplane such that the margin between datasets in the two classes is maximum. In other words, given labeled dataset, SVM outputs the optimal hyperplane that gives the largest minimum distance to the training examples. Formally, the optimization formula is given by [8]:

$$min_{\omega,b} L(\omega) = \tfrac{1}{2}|\omega|^2$$

subject to $y_i(\omega^T x_i + b) \geq 1 \ \forall i$

2.4.1 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) provides a learning technique, used famously in neural network learning [9], also works well for SVM. The goal is to find the optimum value of ω that minimizes the cost function. To minimize this cost function, an iterative approach is used [10],

$$\omega \leftarrow \omega - \eta(\alpha\frac{\partial R(\omega)}{\partial \omega} + \frac{\partial L(\omega^T x_i + b, y_i)}{\partial \omega})$$

where η is the learning rate which controls the step-size in the parameter space, $R(\omega)$ is the regularization term and $L(\omega, b)$ is the loss function. The intercept b is updated similarly but without regularization. The learning rate is adjusted in each iteration and is given by,

$$\eta^{(t)} = \frac{1}{\alpha(t_0+t)}$$

where t is the time step.

3. Methodology

3.1 Data Acquisition

The web archives of The Kathmandu Post [1] were used for the analysis. More than a decade of news (Jan, 2004 to October, 2017) were extracted from the website, using scraping tools. A total of 174,501 news articles. The titles of these news articles, the full text and the corresponding published dates were scraped and saved in a csv file to be later parsed and used as the input data to the trained classifier.

3.2 Training of models

3.2.1 Training Dataset

News articles originating from BBC News, that were rigorously labelled into categories, and made available

by [11], were used to train the model.

The dataset consists of 2,225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005. They are divided into five categories as:

- Business
- Politics
- Tech
- Entertainment
- Sports

For the purpose of this research, the news from the third category (tech) were labeled as class '1' and rest of the news were labeled as class '0'. This way, the problem became a binary classification problem. The full news texts were used for training the classifier.

The dataset, once vectorized, were fed as training dataset to the Naive Bayes' and Support Vector Machine Models.

3.3 Tuning hyper parameters

Hyper-parameters are the parameters that have to be supplied manually, they are not learned by the estimator itself. To tune these parameters to their optimal values, grid search was used, which is an exhaustive search that finds the hyperparameters corresponding to the best cross validation score [12] . All the valid combination of the parameters were applied and the validation score measured for each, the one with the best score was returned.

A valid parameter space for each of the classifiers were fed and grid searches, with the training dataset were carried out. where,

max_df: When building the vocabulary, ignore terms that have a document frequency strictly higher than the given threshold (corpuswise stopwords)

ngram_range: The lower and upper boundary of the range of n_values for different n_grams to be extracted

alpha: Additive (Laplace or Lidstone) smoothing parameter (0 for no smoothing).

norm: Normalization. cosine when norm='l2'

Table 1: Optimal parameters given by Grid Search

	Naive Bayes	SVM
Vectorizer	max_df: 0.5, ngram_range:(1,2)	max_df: 0.75, ngram_range:(1,2)
TF-IDF	norm: l2	norm: l2
Classifier	alpha: 1e-06	alpha: 1e-05, penalty: elasticnet
Best score	0.983	0.992

Table 2: NB: Before and after Grid search

	Default	Param tuning
Training time	0.890452	2.489842
Classification time	0.184031	0.25813
Accuracy	0.905618	0.988764

penalty: The penalty (aka regularization term) to be used. Defaults to 'l2' which is the standard regularizer for linear SVM models. 'l1' and 'elasticnet' might bring sparsity to the model not achievable with 'l2'.

The tuned parameters were then used as the input parameters for the classifiers [12] .

The resulting classified data were used as input to a linear regression model of degree 2 which was used to build the projection for the trend of coverage of technological news for the next 5 years. The linear regression was performed by RANSAC (Random Sample Consensus) algorithm [13].

4. Results and Discussion

4.1 Test set prediction accuracy

To test the accuracy of the classifiers, the BBC News data set was split up into 2 parts in the ratio 4:1, one was used as training data and the other as test data. The trained model was used to predict the classes of the test data. The results for each of the Naive Bayes classifier and SVM are discussed in tables 2 and 3 respectively.

As is evident, SVM gave better accuracy, though by a very small percentage. For this reason, SVM was used to classify the news data from The Kathmandu Post for trend analysis.

Table 3: SVM: Before and after Grid search

	Default	Param tuning
Training time	0.76695	2.584721
Classification time	0.148993	0.293818
Accuracy	0.988764	0.993258

Table 4: Sample Confusion Matrix

N = 360 Samples (Random)	Predicted: Positive	Predicted: Negative
Actual: Positive	TP = 54	FN = 2
Actual: Negative	FP = 16	TN = 288

True Positive Rate: 77.14%
False Positive Rate: 22.85%
True Negative Rate: 99.31%
False Negative Rate: 0.69%
Overall accuracy: 95%

4.2 Train set prediction accuracy

The news articles obtained from the data collection phase were fed to the SVM classifier which categorized them into either of the two categories: Technology and Others. A random sample (n = 360) of the predicted data was taken and the true class were determined manually. The full text of each of the news article from the sample was read by two human readers who analyzed the text and determined if it was tech-related or not (the categories by both the readers matched). Finally, this manual classification was compared against the classifier's classification. The results are shown in Table 4.

From the statistical analysis of the sample, it can be observed that the non-tech news were classified with high accuracy, while the tech news classification has suffered a slight setback. One prominent reason for this, as observed while manually classifying the data, was that the news contained some tech terms (for example: Facebook, Google, telecom, etc.) even though the article as a whole was not tech-related. Other reasons could be the mismatch in contents of the training dataset (BBC News) and the english-language Nepali newspaper (The Kathmandu Post).

4.3 Trend Analysis

With the news items classified, the trend of penetration of technology was visualized in a time series, starting from year 2004, going all the way to 2017. As can be seen from figure 1, the number of tech news has seen an increase since 2003, increasing by a small percentage every year upto 2009. At 2010, it saw a drastic increase, and has been going up and down slightly since. Likewise, from the figure 2, where the number of tech news relative to the total news has been graphed, it can be observed that the trend is increasing as well, peaking at 2016 (Since 2017 is lacking data from 2 months - Nov and Dec, the number seems to be lower). However, even at the highest (i.e, at 2016), technology news has covered less that 3.5% of the total news share.

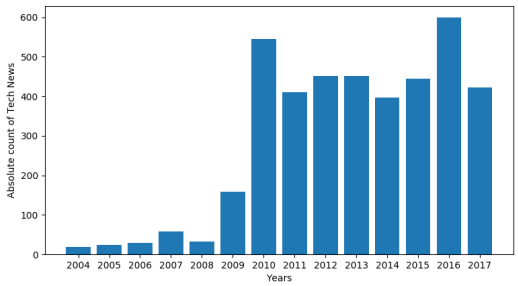

Figure 1: Number of Tech news for every year

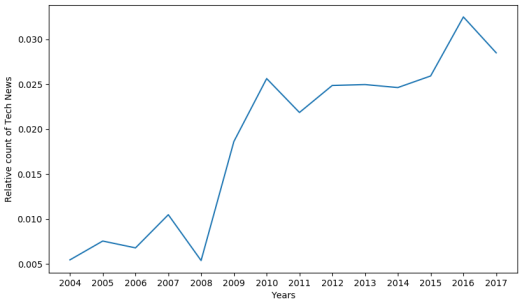

Figure 2: Relative number of Tech news for every year

This observation indicates that tech penetration has still been lagging in the Nepali news scene.

Finally, the time series obtained from the classification was used to make a primitive prediction of tech-related news coverage in the future. Figure 3 shows that it is going to increase, both the absolute number and the relative number, albeit at a slow rate.

5. Conclusion and Future Work

In summary, this research studied the coverage of technology in Nepali news throughout 2004 to 2017 and

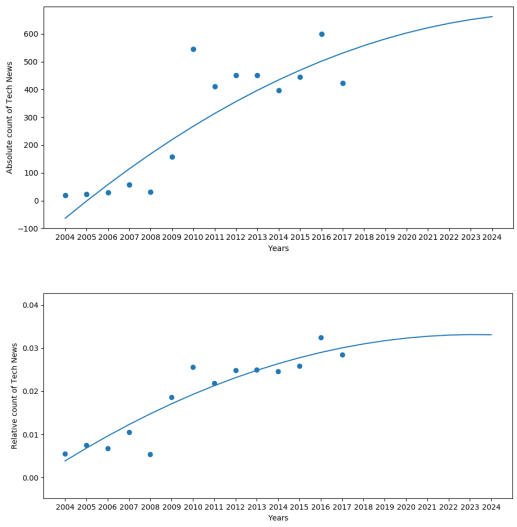

Figure 3: Projection of absolute (upper) and relative (lower) Tech coverage for 5 future years

offered a prediction for the growth of coverage of technology related news in the future. The said coverage was found to have drastically increased from 2008 to 2010 and has been on a rising trend, peaking at 2016. Also, it has been predicted to grow steadily through the coming years. The news classification was done using Support Vector Machines with Stochastic Gradient Descent, which was found to give better accuracy than Naive Bayes' classifier.

There are places that could use some improvements. For instance, instead of using a foreign news dataset for training the learning models, a local news dataset (if was available) could result in better accuracy in classification. This research also opens doors for future substantial research, such as analysing the trend of multiple news platform to get a view of tech penetration in the overall Nepali news industry and analyzing the said trend in relation to different sectors like academia, business or politics (multiclass classification).

Acknowledgments

The authors would like to thank Mr. Bibek Dahal for insights and comments that greatly improved the research as a whole.

References

[1] The kathmandu post archive. `kathmandupost.ekantipur.com/archive`. [Online].

[2] Dewi Y. Liliana, Agung Hardianto, and M. Ridok. Indonesian news classification using support vector machine. 2011.

[3] Neeru Sharma and Paramjit Kaur. Categorize online news using various classification techniques. 2015.

[4] Feature extraction. `en.wikipedia.org/wiki/Feature_extraction`, 2017. [Online; accessed 25-Oct-2017].

[5] Anand Rajaraman and Jeffrey Ullman. *Mining of Massive Datasets*. 2nd edition, 2011.

[6] Yohei SEKI, Hitotsubashi Chiyoda-ku, and Chitosedai Setagaya-ku. Sentence extraction by tf-idf and position weighting from newspaper articles. In *Proceedings of the Third NTCIR Workshop*, 2002.

[7] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. 1998.

[8] Andrew Zisserman. The svm classifier. `http://www.robots.ox.ac.uk/~az/lectures/ml/`.

[9] L'eon Bottou. Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nımes*. AT&T Bell Laboratories, Holmdel, NJ 07733 USA, 1991.

[10] Constantinos Panagiotakopoulos and Petroula Tsampouka. The stochastic gradient descent for the primal l1-svm optimization revisited. 2013.

[11] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press, 2006.

[12] Machine learning in python. `http://scikit-learn.org/stable/`. [Online].

[13] Martin Fischler and Robert Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. 1980.