Named-Entity Based Sentiment Analysis of Nepali News Media Texts

Birat Bade Shrestha, Bal Krishna Bal
Information and Language Processing Research Lab,
Department of Computer Science & Engineering,
Kathmandu University, Dhulikhel, Kavre, Nepal
badebirat@gmail.com, bal@ku.edu.np

Abstract

Due to the general availability, relative abundance and wide diversity of opinions, news Media texts are very good sources for sentiment analysis. However, the major challenge with such texts is the difficulty in aligning the expressed opinions to the concerned political leaders as this entails a non-trivial task of named-entity recognition and anaphora resolution. In this work, our primary focus is on developing a Natural Language Processing (NLP) pipeline involving a robust Named-Entity Recognition followed by Anaphora Resolution and then after alignment of the recognized and resolved named-entities, in this case, political leaders to the correct class of opinions as expressed in the texts. We visualize the popularity of the politicians via the time series graph of positive and negative sentiments as an outcome of the pipeline. We have achieved the performance metrics of the individual components of the pipeline as follows: Part of speech tagging – 93.06% (F1-score), Named-Entity Recognition – 86% (F1-score), Anaphora Resolution – 87.45% (Accuracy), Sentiment Analysis – 80.2% (F1-score).

1 Introduction

In recent times, the way the general public acquires news has drastically changed. Traditional news sources such as television, radio and printed newspapers are in a steady decline in terms of use and consumption. Nowadays, most of the people, especially those from the younger generations read the news on the web either from social media or online news portals. As the internet has become more accessible and available to the general public, we have seen a rapid growth in the number of online news portals and blog sites. Almost all of the established media houses have gone online thereby maintaining online news portals besides the hard copy versions. This makes news media texts a very good resource for sentiment analysis in the socio-political domain.

In most countries the popularity of the politicians (especially, the head of state) is tracked by the media houses, independent organizations as well as the party the politicians are affiliated to. The approval ratings show how popular or unpopular a politician is in the view of the general public. Politicians make change to their policies as well as their public persona so that their approval ratings can improve. Positive approval rating can even suggest the likelihood of a politician wining an election. Approval ratings are generally calculated by conducting opinion polls in a particular sample population. Another way of calculating the approval rating can be by finding out what kind of views are being expressed about a politician in printed news media articles. In the context of our country, such approval ratings are not available. This work presents a way of calculating popularity ratings and thus helping to determine how popular or unpopular a Nepali politician is by analyzing the news media texts.

The task of analyzing sentiments in the news media texts has its own set of challenges. A news article may contain sentiments or opinions expressed over more than one politician. Worse, even a sentence might refer to sentiments expressed over multiple politicians. From this perspective, the task demands that the named-entity or the political leader being referred to is accurately identified and resolved in terms of the pronominal references used in the text before moving to the task of aligning the corresponding sentiment to the named-entity. Once the two phase task of named-entity resolution and sentiment alignment is complete, we present the results in the form of a time-series popularity or trending graph. Such a representation would be of interest to a wide range of target audience – the political

Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications, pages 114–120
Suzhou, China, December 4, 2020.

leader himself/herself, his/her political affiliation, the general public and media houses.

Automating the task is not simple or trivial as the whole process is quite technically involved and requires a series of NLP sub-tasks to be accomplished before we finally reach the end of the pipeline. We describe the pipeline in Section 3 of this paper.

In terms of accomplishing the work, our major contributions can be listed as follows:

a) Developed a Part-of-Speech (PoS) tagger for Nepali
b) Developed a Named-Entity Recognition (NER) classifier for Nepali
c) Developed a rule-based Anaphora Resolution module for Nepali
d) Manually labelled a sentence level Sentiment Corpus of 3490 sentences from Nepali News Media texts
e) Developed a Machine Learning based Sentiment Classifier based on the Sentiment Corpus

2 Related Work

There have been a few research works on Sentiment Analysis in Nepali texts. In one of the first works on Sentiment Analysis for the Nepali Language, Gupta and Bal developed a lexical resource namely the Bhavanakos (Gupta & Bal, 2015).

Yadav & Pant (2014) used a machine learning approach to determine if movie reviews were positive or negative. Their architecture consisted of 3 major components, viz., Pre-Processing, Feature Extraction and Classification. In the Pre-Processing phase, they performed the steps such as whitespace and special character removal, abbreviations expansion, stemming, stop word removal, negation handling, PoS tagging, named-entity recognition etc. To train the model they extracted features such as TF-IDF, positive word count, negative word count, presence of polar words etc. Using these parameters they were able to train a Naïve Bayes classifier and reported a precision of 79.23%, a recall of 78.57% and F-score of 78.90%. Their data set consisted of 500 samples: 250 positive and 250 negative (Yadav & Pant, 2014).

In an undertaking similar to sentiment analysis, Shahi & Pant (2018) used different text mining techniques to address the text classification problem for Nepali news media text. The researchers compared the accuracy of three machine algorithms: Naïve Bayes, Neural Network and Support Vector Machine to classify text according to their content. Their architecture consisted of 3 major components: Pre-Processing, Feature Extraction and Machine Learning. In the Pre-Processing phase they performed the steps such as tokenization, special symbol and number removal, stop word removal and word stemming. To extract the feature from the dataset they used TF-IDF. The researchers have used two instances of the SVM: SVM with Linear Kernel and SVM Radial Basis Function kernels. SVM is a binary classifier but the problem of text classification is a multi-class problem. In order to mitigate this issue the researchers adopted a one-vs.-rest approach. The Neural Network they used was a simple dense backpropagation multilayered perceptron with stochastic gradient descent optimization. The researchers used a five-fold validation method. Their result showed that SVM with RBF kernel outperformed the other three algorithms with an average accuracy of 74.65%. Linear SVM had an average accuracy of 74.62%, Multilayer Perceptron Neural Networks had 72.99% and lastly the Naive Bayes had an accuracy of 68.31% (Shahi & Pant, 2018).

Researchers have used sentiment analysis techniques to extract opinion from News Media texts as well. Thapa & Bal (2016) compared the accuracy of Machine Learning algorithms to classify sentiments expressed in Nepali news media text. They tested three algorithms: Support Vector Machine, Multinomial Naive Bayes and Logistic Regression using a 5-fold cross validation method. Their dataset consisted of 384 book and movie reviews. In the dataset, 179 were positive and 205 were negative sentences. To extract the feature from the dataset, they used four methods: Bag-of-Words, Bag-of-Words (with stopwords removed), TF-IDF and TF-IDF (with stopwords removed). The results obtained from their experiments showed that the F1-score of Multinomial Naive Bayes Algorithm was higher when taking TF-IDF (with stopwords) (Thapa & Bal, 2016).

In addition to this, Kafle (2019) implemented a sentiment based popularity tracker for Nepali politicians. In his research, he used Nepali news media text published in English as his data source. He tracked the popularity of Nepali politicians based on two parameters: growing popularity,

diminishing popularity. To track the popularity of a particular politician, he carried out a sentence level sentiment analysis and assigned sentiment scores to each article with respect to that politician. His architecture can be divided into three main phases. The first phase was Named-Entity Extraction where all the named-entities in the articles were extracted. In the second phase, pronomial anaphora were resolved by replacing the pronouns with the named-entities they were referring to. And in the third and final phase sentiments were extracted from the articles tokenized into sentences. To extract the sentiments from the tokens they used a lexicon and rule-based sentiment analysis tool called Valence Aware Dictionary and sEntiment Reasoner (VADER).

Nepali language is a morphologically rich and complex language. In order to mine opinions from Nepali texts, the text classifier being used should be able to incorporate specific language features before classifying the text (Shahi & Pant, 2018). Different techniques have been used in order to extract sentiment from Nepali text. Researchers have employed learning based as well as rule and lexicon based approach for sentiment classification. To extract features from Nepali text, techniques like TF-IDF, Bags-of-Words etc are used. But the problems with these techniques are that they do not consider the context of the words. It has also been observed that removing stop words has no significant effect on the evaluation metrics and the performance of the classifier (Thapa & Bal, 2016).

3 Methodology

We propose the following framework to address the given research problem, which consists of a pipeline of six components:

1. Data Collection
2. Pre-Processing
3. Parts-of-Speech Tagging
4. Named-Entity Recognition
5. Anaphora Resolution
6. Sentiment Analysis

3.1 Data Collection

A multi-threaded scraping framework was developed in order to facilitate the data collection process. The data was gathered by scrapping news articles from four online news portals. The scrapping framework scraped the news articles from online news portals and saved them in a data repository. The news portals were chosen based upon their influence as a mainstream news source in the Nepali news media. The online portals from which the articles were scrapped are:

i. Kantipur Daily [1]
ii. NagarikNews [2]
iii. Online Khabar [3]
iv. Setopati [4]

3.2 Pre-Processing

The Pre-Processing component consists of two sub-components: Article Cleaning and Article Lemmatizing. First of all, badly encoded characters are removed from the articles. The scrapping framework itself is equipped with the functionality to strip the unnecessary HTML tags and JavaScript codes. Unnecessary punctuation symbols are also removed from the articles and the articles are prepared for the next phase by tokenizing them.

Secondly, the Article Lemmatizing sub-component takes care of the removal of certain suffixes at the end of each tokenized word such as **लाई, मा, हरू, को** etc. Need to note that most inflections in the Nepali language are caused by the postpositions that are placed after nouns, verbs etc. The list of post-positions was obtained from sanjaalcorps [5] .The Article Lemmatizing sub-component splits the root word and the suffixes but does not remove the suffix altogether as it is required in the subsequent phase.

3.3 Part-of-Speech Tagging

The Part-of-Speech (PoS) tagging component assigns lexical category to each word in a text. A Part of Speech is a category of words that have similar grammatical properties. The most common PoS for English language are noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection, numeral, article or determiner.

[1] https://ekantipur.com/
[2] https://nagariknews.nagariknetwork.com/
[3] https://www.onlinekhabar.com/
[4] https://www.setopati.com/
[5] https://github.com/sanjaalcorps/

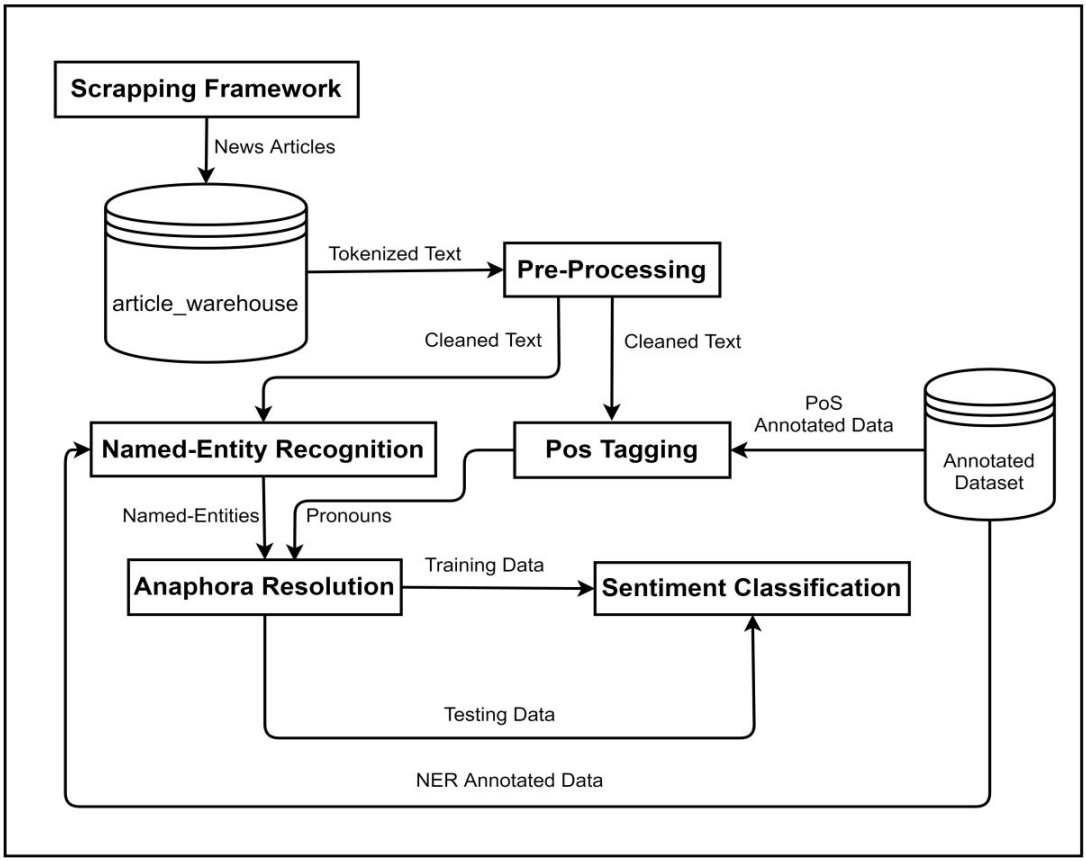

Figure 1: System Architecture

We trained the statistical based Trigrams 'n" Tagger (TnT) tagger with the PoS tagged corpus of Nepali available at the official website of the Center for Language Engineering [6]. The TnT tagger is based on the work of Thorsten Brants (Brants, 2002). If the tagger encounters a new word that has not been trained before, it will tag that word as 'Unk' or Unknown. In order to address the 'Unk' part of speech category, we added a few articles from our corpus into the training dataset.

A 10-fold validation method was used to validate the model. The results showed that the trained model had an average accuracy of 90.05%, precision of 96.55%, recall of 91.3% and F1-score of 93.06. It was also found that lemmatizing the text increased the accuracy of the model by 10%.

3.4 Named-Entity Recognition

A Named-Entity refers to a real-world object such as the name of the person, organization, location etc. In Named-Entity Recognition, such named entities are identified and tagged in text. A Named-Entity Recognition classifier reads the texts, identifies the named entities and classifies them accordingly. For the NER component, we train the StandardNERTagger. This NER classifier is based on the work of Finkel et al (Finkel et al., 2005). It uses an advanced statistical learning algorithm and therefore is relatively computationally expensive. For this work, we used the dataset made available by (Singh et al., 2019). The dataset follows the standard CoNLL-2003 IO format (Sang and Meulder, 2003). The dataset consists of two tab separated fields: the word, named entity tag with one word per line. In order to enhance the performance of the tagger, we extended the dataset with data from our own corpus. We again used a 10-fold validation method to validate the model. The model had an average F1-score of 86%. We present the Precision, Recall and F1-scores for the named-entity tags in Table no 1.

[6] http://www.cle.org.pk/

Tag	Precision	Recall	F1-score
PER	97	85	90
LOC	89	66	76
ORG	87	74	80
O	97	99	98

Table 1: Performance metrics of NER

3.5 Anaphora Resolution

Anaphora resolution refers to the task of correctly resolving the pronominal references in terms of the referred Named-Entity in texts. The referring word is called anaphora and the referenced word is called antecedent.

We implemented a rule-based method based on the Lappin and Leass algorithm to resolve the anaphora on Nepali news media text. The algorithm uses a simple weighting scheme that balances the effects of recency and sentence structure. The algorithm works by adding a discourse variable for each new entity mentioned in the discourse. The algorithm calculates the degree of salience for each entity by summing the weights from a table of salience factors (Lappin and Leass, 1994).

This component uses the pronouns and other parts of speech tagged by the PoS tagging component and the Named-Entities tagged by Named-Entity Recognition Component. In this research work, we resolve the personal pronouns such as **उनी, उन, उहाँ, उहा, उ, ऊ, वहाँ** only. For testing, the first five sentences of 292 news articles were used thus making it a total of 1460 sentences. Out of the 1460, 600 sentences had no named entities whereas 589 had named entities in them and 271 sentences had pronouns. Out of the 271 pronouns, 237 were resolved correctly and 34 were resolved incorrectly. The accuracy of the algorithm was found to be 87.45 %.

3.6 Sentiment Analysis

In this section, we discuss how the sentiment analysis model was developed.

3.6.1 Dataset

Unfortunately, a publicly available sentence level sentiment annotated dataset for Nepali language is not available. Most previous research deals with document level sentiment analysis so they are of very little use for this work. The only option that remained was to manually label the dataset. A total of 3490 sentences were labeled manually. The sentences were classified into one of the following two classes; Positive and Negative. Out of these 3490 sentences, 2676 were positive sentences and 814 were negative. To make the dataset balanced, we included an equal number of positive (814) and negative (814) sentences in the training dataset.

3.6.2 Word Embedding

In Natural Language Processing, representing words in multi-dimensional vector space can improve the performance of a learning based algorithm (Mikolov et al., 2013). In this work, we have used Word2Vec (Mikolov et al., 2013) and FastText (Bojanowski et al., 2016) to embed the words primarily for the purpose of feature extraction for the next component in the pipeline, i.e., Sentiment Classification. There exist pre-trained embedding models for the Nepali language texts (Lamsal, 2019). However, to ensure better coverage, we went for embedding models based on text corpus developed for this research work. The trained model represents each word from the corpus in a 300 dimensional vector space thereby making it 19339 unique words. Four models, Word2Vec with CBOW, Word2Vec with skipgram, FastText with CBOW and FastText with skipgram were trained for testing purposes. To extract the necessary feature vector, firstly the words in the sentences were embedded, and then we averaged the embedding vector of each word in a given sentence. Finally this feature vector was used for sentiment classification. ,

From the initial experimentations, we found out that the classifiers were performing better when the words were embedded using the skipgram parameter. The results obtained are presented in Table no 2.

Model	Parameter	F1
Word2Vec	Continuous Bag of Words	70
	Skip Gram	**80.2**
FastText	Continuous Bag of Words	66.4
	Skip Gram	**78.7**

Table 2: Performance Evaluation of Embedding Methods

3.6.3 Sentiment Classification

For classifying the sentiments in this work, the initial plan was to use Recurrent Neural Network

specialized for Natural Language Processing but due to the small size of our dataset, we used three other machine learning algorithms: Support Vector Machine, Decision Tree and Random Forest.

4 Results

In order to test the performance of the sentiment classifiers, 10-fold cross validation method was used. From the experiments conducted, we found that Support Vector Machine (SVM) with Word2vec embedding (skipgram) had the overall highest performance metrics with an accuracy of 80.15%, precision of 80.4%, recall of 80.2% and F1-score of 80.2%. All the averaged results obtained from the experiments are presented in Table no 3.

Algorithm	**Embedding**	**F1-score**
SVM	**Word2Vec**	**80.2**
	FastText	78.7
Random Forrest	Word2Vec	77.1
	FastText	76.6
Decision Tree	Word2Vec	68.2
	FastText	67.9

Table 3: Performance evaluation of SVM, Random Forrest and Decision Tree

For visualization, popularity trends of three most prominent Nepali politicians from 2018/04 to 2018/09 were plotted. The popularity graphs are shown in Figure no 2.

5 Conclusion

We implemented a named-entity based sentiment analysis framework in this research work in order to distill the outlook expressed towards the politicians in the news media. Initially, the names of the politicians in the news articles were identified and the pronominal expressions that were referring to the names were resolved. Sentences with the name of the politicians were then extracted and classified according to the sentiment expressed towards them.

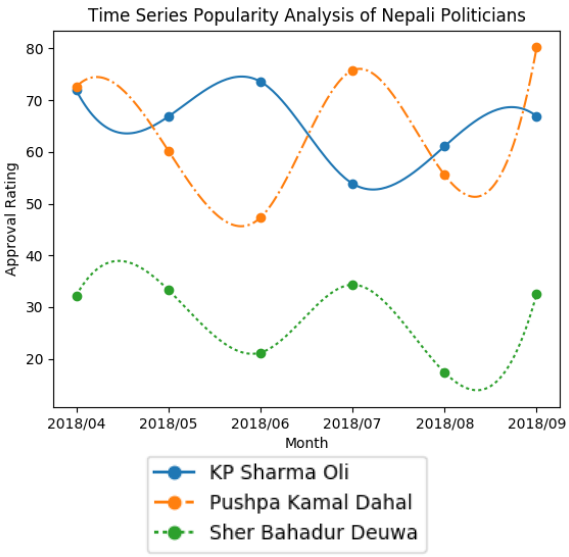

Figure 2: Time Series Popularity Analysis of Nepali Politicians

Support Vector Machine had the overall highest performance metrics for classifying the sentiments expressed in the sentences. We experimented with different embedding techniques and found that Word2Vec with skipgram was the optimal option for feature extraction. This combination of classifier and embedding technique was used to classify the sentences in the articles. Finally, as an application of our research work, we presented the results in the form of a time-series popularity graph or a trending graph.

For each component of the pipeline, we were able to achieve a relatively higher performance metric. Nevertheless, there are some limitations to our work. The components of the proposed framework are based on probabilistic and rule based models, which underperform compared to the neural network models. We could not go for the latter because our dataset is not large enough. Similarly, the anaphora resolution component only resolved pronouns. Other expressions referring to an entity were ignored all together. Furthermore, we have not dealt with the opinion holder or the target explicitly in terms of opinions in this work although named-entities can also be related to the target in many aspects.

The performance of the overall framework can be further enhanced by using more advanced variants of Neural Networks, which are specialized for Natural Language Processing tasks. Different variants of RNN have been used

for Parts-of-Speech tagging as well as Named-Entity Extraction with considerable accuracy and success for other languages. These Neural Networks can be used for sentiment classification as well, given we have sufficient data. So, one aspect of future enhancement to the work would definitely be increasing the dataset. There are other areas of improvements to this work. For anaphora resolution, Graph Based Neural Networks seem to be better as graph based neural networks are more effective in handling non-linear data. Furthermore, latest embedding techniques such as BERT, ELMO, etc. can be used to more accurately embed the context of words within a sentence and thus get better results.

References

Abhimanu Yadav and Ashok K. Pant. 2014. Sentiment Analysis on Nepali Movie Reviews using Machine Learning. *Journal for Research and Development.*

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114-133. https://doi.org/10.1145/322234.32224.

Chandan Gupta and Bal Krishna Bal. 2015. Detecting Sentiment in Nepali texts: A bootstrap approach for Sentiment Analysis of texts in the Nepali language. 1-4. 10.1109/CCIP.2015.7100739.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363-370.

Kamal Kafle. 2019. Popularity Tracking and Trend Analyses of Political Figures Based on Online News Data (Master's thesis). Kathmandu University

Lal Bahadur Reshmi Thapa and Bal Krishna Bal. 2016. Classifying sentiments in Nepali subjective texts. 1-6. 10.1109/IISA.2016.7785374.

Oyesh Mann Singh, Ankur Padia, and Anupam Joshi. 2019. Named Entity Recognition for Nepali Language.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprintarXiv:1607.04606*.

Rabindra Lamsal. 2019. 300-Dimensional Word Embeddings for Nepali Language. IEEE Dataport.http://dx.doi.org/10.21227/dz6s-my90

Shalom Lappin, Herbert J. Leass. 1994. An algorithm for pronomial anaphora resolution. Computational Linguistics 20, 535–561

Tej Bahadur Shahi and Ashok Kumar Pant. 2018. Nepali news classification using Naïve Bayes, Support Vector Machines and Neural Networks. 1-5. 10.1109/ICCICT.2018.8325883.

Thorsten Brants. 2002. TnT: A Statistical Part-of-Speech Tagger. ANLP. 10.3115/974147.974178.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.