南京信息工程大学

Advanced Artificial Intelligence

Title: Natural Language Processing with Naive Bayes, Decision Tree, and Random Forest

Student Name: Musaazi Godfrey

Student ID: 20175243014

Major: Computer Science and Technology

Submitted To

School of Computer and Software

Nanjing University of Information Science & Technology

20th December, 2018

Supervised By: Le Sun (孙乐), Ph.D.

Contents

Natural Language Processing with Naive Bayes, Decision Tree, and Random Forest

Musaazi Godfrey

Abstract

Leading companies are leveraging the benefits of NLP in sentiment analysis of customers for better marketing, and improving service delivery. Companies can identify the sentiment of customers' feedback into positive, negative, or neutral categories. Natural Language Understanding (NLU) which is a branch of NLP is a hard task, choosing the best classifier while building an NLU based algorithm is one of the challenges at hand. In this work, we evaluate the performance of Naïve Bayes, Decision Tree, and Random Forest classifiers in handling sentimental analysis problem. Our experimental results show that Naïve Bayes achieves 90% accuracy, Decision tree achieves 90.5%, and Random Forest achieves 85% accuracy. If we consider only accuracy, DT becomes our best model but accuracy is not enough and therefore other performance metrics like precision, recall and F1 score must be considered as well. To achieve the objective of this paper, we computed a full score by summing accuracy, and F1 score. Full score was 1.664 for Naïve Bayes, 1.628 for DT, and 1.476 for Random Forest. Naïve Bayes with 90% accuracy and 0.764 F1 score was the best algorithm for our case study.

Keywords: Natural language processing, natural language understanding, sentimental analysis, naïve bayes, decision tree, random forest.

1. Introduction

Natural language processing (NLP) is a subfield of artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. Challenges in natural language processing frequently involve speech recognition, natural language understanding, and natural language generation (Natural_language_processing). NLP has been applied in many fields like Sentiment analysis, Customer Service, banking, healthcare, and automotive. For example, Nina (Intelligent automated virtual assistant technology), is an intelligent automated virtual assistant technology for customer self-service, it engages customers in natural conversations using voice or text.

Natural-language understanding (NLU) or natural-language interpretation (NLI) is a subtopic of natural-language processing in artificial intelligence that deals with machine reading comprehension. Natural-language understanding is considered an AI-hard problem. NLU has been applied to question answering, news-gathering, text categorization, voice-activation, archiving, and large-scale content analysis (Natural-language_understanding).NLU through sentiment analysis of customers is utilized by top companies to know what people are saying, how they're saying it, and what they mean. Companies are able to identify the sentiment of customers' feedback into positive, negative, or neutral categories. Data analysts in large companies do market research, brand monitoring, investigate product reputation and understand customer experiences by using sentiment analysis.

Machine learning techniques through Supervised learning, Semi-supervised and Unsupervised methods have been applied to sentimental analysis classification challenges and have proved promising results. In this paper, our focus is on supervised learning methods. They depend on the existence of labeled training documents. Supervised learning is a successful solution in classification and has been used for sentiment classification with very promising results. Some of the most frequently used supervised classification methods in sentiment analysis are SVM, NB Maximum Entropy (ME), Artificial Neural Network (NN) and Decision Tree (DT) classifiers. Some other less commonly used algorithms are LR, K Nearest Neighbor (KNN), RF and Bayesian Network (BN) (Aydogan & Ali Akcayol, 2016). For most of the works reviewed, researchers have comparatively used at least two classifiers in solving a sentiment analysis classification problem. (M.Cetin and M.F.Amasyali, 2013) used NB, DT, RF and KNN algorithms comparatively in Turkish sentiment analysis. Their results showed that better test success ratio can be obtained with fewer training data selected by active learning than using all training set with NB classifier. (M.Meral and B.Diri, 2014) carried out Turkish sentiment analysis on Twitter data using NB, SVM and RF algorithms. (P.C.R.Lange, D.Clarke, and P.Hender, 2012), (M.Cetin and M.F.Amasyali, 2013) (Kaynar, Aydin, & Görmez, 2017), (M.Meral and B.Diri, 2014), (S.E.Seker, K.Al-Naami, 2013), (B.Florian, F.Schultze, and L.Strauch, 2015), (P.H.Shahana, B.Omman, 2015) employed DT in sentiment analysis.

2. Fundamentals

2.1 Naïve Bayes

Naïve Bayes is a simple but powerful and commonly used probabilistic machine learning classifier. The mathematical concept behind it is Bayes theorem. It makes classifications using the highest posterior probability.

Consider classes $a_1, a_2, \ldots, a_m$. For a new data point X with features $x_1, x_2, \ldots, x_n$ to be classified among classes $a_1, a_2, \ldots, a_m$, Naïve Bayes classifier determines the probability of the features occurring in each class and basing on the highest probability, it returns the most likely class. For each class it calculates $p(a_i \mid x_1, x_2, \ldots, x_n)$ $for\ i = 1 \ldots m$ using Bayesian rule shown in figure 2 below.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Figure 1: Bayesian Rule

2.2 Decision Tree

A decision tree is a hierarchical data structure implementing a divide-and-conquer strategy. It is an efficient nonparametric method, which can be used both for regression and classification. It is a supervised machine learning model that performs recursive splits in a smaller number of steps to determine the local region. A decision tree has internal decision nodes and terminal leaves (see figure 3). Each decision node m, implements a test function $fm(x)$ with discrete outcomes labeling the branches (Alpaydin, 2004).For an input X, at each node, a test is applied and one of the braches is taken depending on the outcome. This process starts at the root and is repeated recursively until a leaf node is hit, at which point the value written in the leaf constitutes the output.

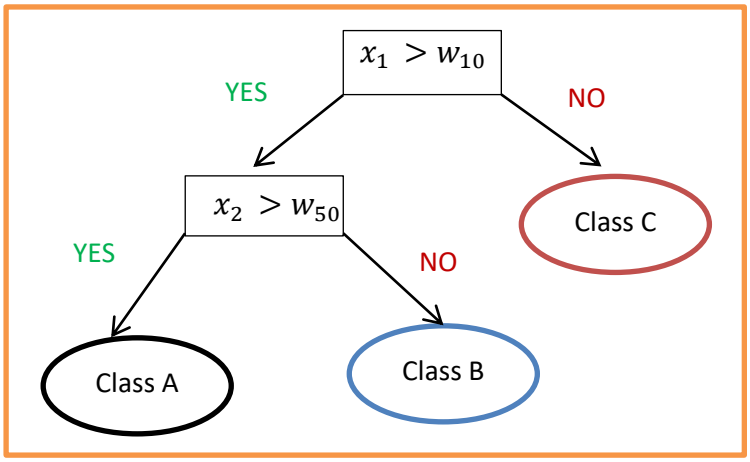

Figure 2: Decision Tree

2.3 Random Forest

A Random forest is a type of ensemble learning algorithm that builds a multitude of decision trees at training time and outputs the class that has the majority vote from all classes predicted by individual trees.

Pseudo code:

1. Pick at random with replacement M data points from a training set
2. Build a decision tree associated to these M data points.
3. Choose the number of the trees N you want to build and repeat steps 1 & 2.
4. For a new data point, make each one of the N trees to predict the class to which the data point belongs, and assign the new data point to the class that wins the majority vote.

A single tree is very sensitive and any change in the dataset can affect its result. Taking the mode of predicted classes by individual trees improves the accuracy of the final prediction by random forest and ensemble algorithms are considered more stable.

2.4 Dataset

The dataset used for this study contains text reviews made by customers who visited a given restaurant. Each review is a string of characters with a label of "0" for customers who didn't like the restaurant and "1" for those who liked the restaurant. There are 1000 observations in total to support our study. 75% of the dataset is used to train models and 25% is used for testing models.

3. Approach

3.1 Sentimental Analysis Process

We use Natural Language toolkit (NLTK) (Project, n.d.) to perform sentiment analysis of our data and we solve the challenge in three stages as shown in figure 1 below.

Firstly, data preprocessing is performed on the dataset. For each review, it involves (1)cleaning text; here we only consider alphabetic characters (a-z and A-Z) in other words we remove punctuation and numbers, (2)converting text to single case (i.e. lowercase), (3)splitting a review into a list of words, (4)removing non-significant words such as "the", "a", "an", "in" (stop words) (Removing stop words with NLTK in Python), (5)considering root words only (stemming) (Python | Stemming words with NLTK); a stemming algorithm reduces the words "liked", "likely", "likes", "liking" to a root word "like" ,(6)converting review list back to single string, and (7)appending the review to a corpus.

In stage two, we generate a matrix of independent features (sparse matrix) using CountVecrtorizer (sklearn.feature_extraction.text.CountVectorizer) which returns the most frequent words in the corpus. We set maximum number of features to 1500 and create a bag of words model, it takes each word(unique, no duplicates) from corpus and creates a column for it hence creating a table with 1000 rows(each row for a review).Each cell, will contain a number representing the number of times the column word appears in the review.

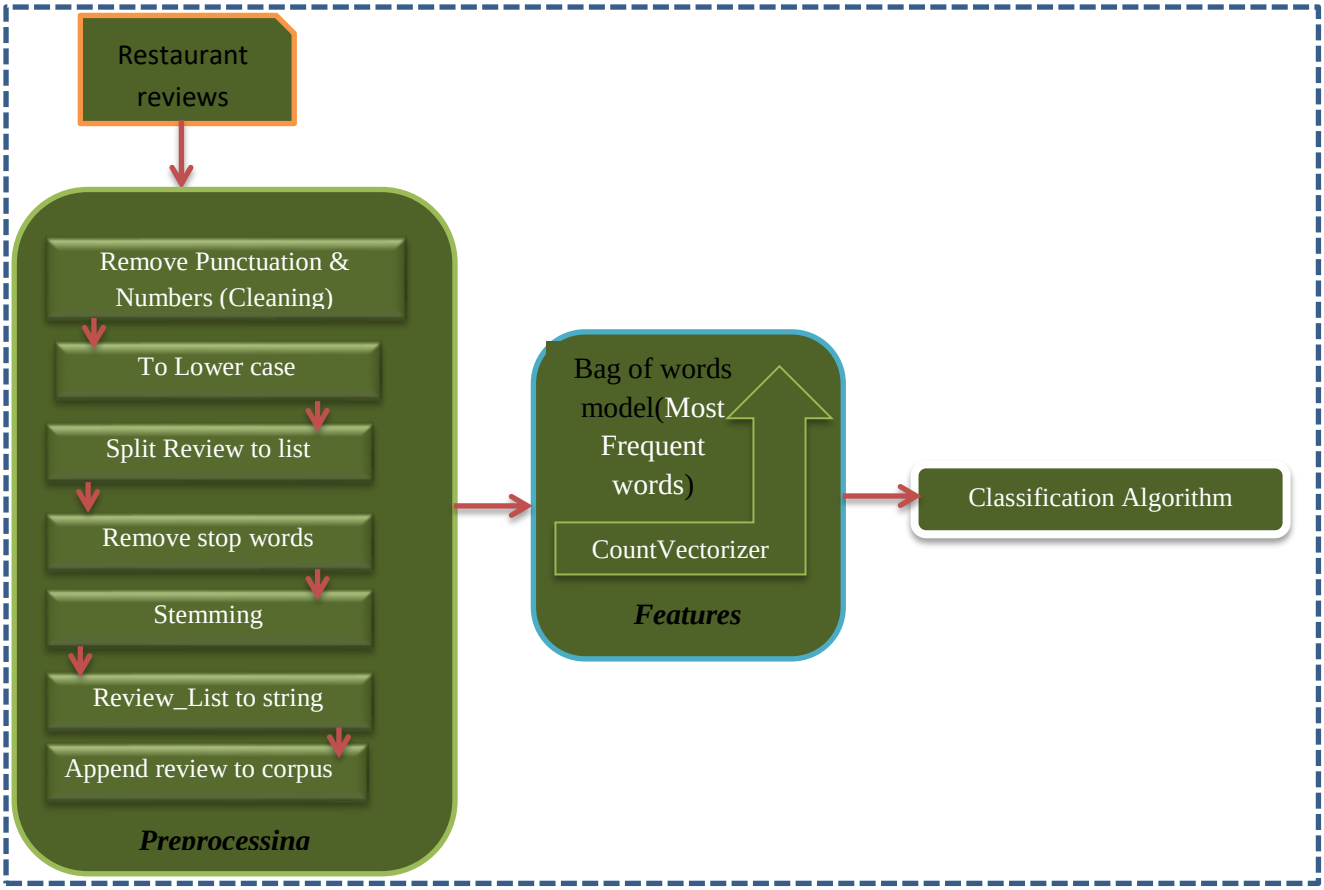

Figure 3: System framework

In the final stage, we build 3 models that are used to classify the reviews comparatively. Model 1 is Naïve Bayes based, Model 2 is based on Decision Tree, and Model 3 is Random Forest based.

4. Experimental Results and Discussion.

Same dataset and *train: test ratio* is used to train and test the 3 models under comparison. We compute accuracy AC as shown in Equation1.

Equation 1: Accuracy

$$AC = \frac{(TP+TN)}{(TP+TN+FP+FN}$$

4.1 Naïve Bayes

Experimental results show that Naïve Bayes achieves 90% accuracy.

	Didn't Like	Like
Didn't Like	60	50
Like	20	113

Figure 4: Confusion Matrix for Naive Bayes

4.2 Decision tree

Decision tree achieved 90.5% accuracy which is slightly better than that for Naïve Bayes.

	Didn't Like	**Like**
Didn't Like	91	26
Like	43	90

Figure 5: Confusion Matrix for Decision Tree

4.3 Random Forest

Random forest performed least with an accuracy of 85%.

	Didn't Like	**Like**
Didn't Like	103	14
Like	66	67

Figure 6: Confusion Matrix for Random Forest

For now if we only consider accuracy as the only metric to compare our models. Decision tree which achieved 90.5% accuracy emerges as the best model. However, accuracy alone is not enough to evaluate model's performance. We therefore evaluate the performance of each of these models using other performance metrics like Precision; which measures exactness, Recall; which measures completeness, and F1 Score; compromise between Precision and Recall.

To compute Precision, Recall, and F1 Score, we use equations 1, 2, and 3 below respectively. Where "TP" implies True Positives, "TN" implies True Negatives, "FP" implies False Positives, and "FN" implies False Negatives.

Equation 2: Precision

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

Equation 3: Recall

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

Equation 4: F1 Score

$$F1\ Score = \frac{2*Precision*Recall}{Precision+Recall}$$

For comparison purposes, we compute $total\ score = accuracy + F1\ Score$

Table 1: Model Performance Evaluation

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Total Score**
Naïve Bayes	0.9	0.85	0.693	0.764	1.664
Decision Tree	0.905	0.677	0.776	0.723	1.628
Random Forest	0.85	0.504	0.827	0.626	1.476

Therefore for this case study, according to results in table 1 above we see that Naive Bayes with 0.9 accuracy and 0.764 F1 Score, emerges as the best out of the 3 algorithms used.

5. Conclusion

In this paper, we have demonstrated how to solve a sentiment analysis problem using natural language tool kit and different algorithms for classification. Natural language Understanding being a hard task for artificial intelligence systems. It is of great value while solving a natural language understanding challenge to test with different algorithms and also use several metrics to evaluate the performance of the built models.

References

B.Florian, F.Schultze, and L.Strauch. (2015). Semantic Search:Sematic Analysis with Machine Learning Algorithms on German News Articles.

Intelligent automated virtual assistant technology. (n.d.). Retrieved November 04, 2018, from NUANCE: https://www.nuance.com/omni-channel-customer-engagement/digital/virtual-assistant/nina.html

M.Cetin and M.F.Amasyali. (2013). Active learning for Turkish sentiment analysis. IEEE International Symposium on Innovation in Intelligent System and Applications(INISTA), (pp. 1-4).

M.Meral and B.Diri. (2014). Sentiment analysis on Twitter. Signal Processing and Communications Applications Conference (SIU), (pp. 690-693).

Natural_language_processing. (n.d.). Retrieved November 04, 2018, from wikipedia: https://en.wikipedia.org/wiki/Natural_language_processing

Natural-language_understanding. (n.d.). Retrieved November 03, 2018, from wikipedia: https://en.wikipedia.org/wiki/Natural-language_understanding

P.C.R.Lange, D.Clarke, and P.Hender. (2012). On developing robust models for favourability analysis:Model choice, feature sets and imbalanced data. Decision Support Systems, vol.53 no.4, (pp. 712-718).

P.H.Shahana, B.Omman. (2015). Evaluation of Features on Sentimental Analysis. Procedia Computer Science, vol.46, no.Icict 2014, (pp. 1585-1592).

Python | Stemming words with NLTK. (n.d.). Retrieved 12 16, 2018, from geeksforgeeks: https://www.geeksforgeeks.org/python-stemming-words-with-nltk/

Removing stop words with NLTK in Python. (n.d.). Retrieved 12 16, 2018, from geeksforgeeks: https://www.geeksforgeeks.org/removing-stop-words-nltk-python/

S.E.Seker, K.Al-Naami. (2013). Sentimental Analysis on Turkish Blogs via Ensemble Classifier. Proceedings in the International Conference on Data Minining.

sklearn.feature_extraction.text.CountVectorizer. (n.d.). Retrieved 12 02, 2018, from scikit-learn.org: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

Alpaydin, E. (2004). Introduction to Machine Learning - Ethem Alpaydin.pdf.

Aydogan, E., & Ali Akcayol, M. (2016). A comprehensive survey for sentiment analysis tasks using machine learning techniques. *Proceedings of the 2016 International Symposium on INnovations in Intelligent SysTems and Applications, INISTA 2016*. https://doi.org/10.1109/INISTA.2016.7571856

Kaynar, O., Aydin, Z., & Görmez, Y. (2017). Sentiment Analizinde Öznitelik Düşürme Yöntemlerinin Oto Kodlayıcılı Derin Öğrenme Makinaları ile K arşılaştırılması Comparison of Feature Reduction Methods with Deep Autoencoder Machine Learning in Sentiment Analysis, 319–326. https://doi.org/10.17671/gazibtd.331046

Project, N. (n.d.). Natural Language Toolkit. Retrieved 18 December 2018, from https://www.nltk.org/