Subjectivity Classification of Filipino Text with Features based on Term Frequency – Inverse Document Frequency

Regalado, Ralph Vincent J., Chua, Jenina L., Co, Justin L., Tiam-Lee, Thomas James Z.
Center for Language Technologies
De La Salle University
Manila, Philippines
ralph.regalado@delasalle.ph, jenina_chua@dlsu.ph, justin_co@dlsu.ph, thomas_tiam-lee@dlsu.ph

Abstract— **Subjectivity classification classifies a given document if it contains subjective information or not, or identifies which portions of the document are subjective. This research reports a machine learning approach on document-level and sentence-level subjectivity classification of Filipino texts using existing machine learning algorithms such as C4.5, Naïve Bayes, k-Nearest Neighbor, and Support Vector Machine. For the document-level classification, result shows that Support Vector Machines gave the best result with 95.06% accuracy. While for the sentence-level classification, Naïve Baves gave the best result with 58.75% accuracy.**

Keywords- subjectivity classification; machine learning approach; Filipino language; TF-IDF

I. Introduction

The advent of the World Wide Web has enabled an increasing amount of users to express their personal experiences and opinions openly through reviews, posts and comments, forums, and blogs, which are called user-generated-content. This online word-of-mouth can contain valuable information and can have practical applications for both consumers and large organization. Consumers can utilize this content to gain first-hand information on a product, expanding their sources from their peers to compete strangers on the internet. Organizations can use these contents to know how their products “stack up” against their competitors without the need to invest on surveys, focus groups, and consultants [1].

Textual information from any context can be classified into two groups: facts and opinions. Facts are user expressions that contain objective information about an entity and their features. Opinions on the other hand are subjective expressions that contain the user sentiment, personal experiences, and emotion towards an entity and its attributes [1]. Although most information are already available on the web, the number of these information are too great to be collected manually, and it is seldom that a consumer or a company would rely on a single opinion or review to dictate the opinion of the majority, but instead they rely on the consensus opinion of all available review on the web. This task of identifying opinionated text and classifying them into the proper polarity group is called sentiment analysis [2].

One of the problems of sentiment analysis is subjectivity classification. Subjectivity classification classifies a given document if it contains subjective information or not, or identifies which portions of the document are subjective [1]. Subjectivity classification is deemed as important as it’s the first process needed to be accomplished before performing polarity classification. Subjectivity classification can be further categorized into two. The document-level subjectivity classification, which classifies if a given document is subjective or not, once classified as subjective it will then proceed to the next classification which is the sentence-level subjectivity classification. Sentence-level subjectivity classification identifies which sentences are subjective or not.

In this paper, a machine learning approach using machine learning algorithms such as the C4.5, the Naïve Bayes, the k-NN, and support vector machines, with words (from the dataset) based on their corresponding Term Frequency – Inverse Document Frequency (TF-IDF) scores of as features in classifying documents written in Filipino as either factual or opinion-based will be discussed. Section 2 reviews existing works related to our approaches. Section 3 introduces the main processes of our approach. Section 4 describes our experiments and findings. In Section 5, we conclude our efforts and discuss some future works.

II. Related Works

The work of [3] was one of the early works that quantify sentiment analysis as a machine learning problem. The work evaluated the use of machine learning by training a dataset derived from a set of annotated documents using simple data representation on established machine learning algorithms. Several feature extraction techniques were used in their approach which includes the use of term frequency, term presence, part-of-speech tags and position of the word. Our work only focuses on one technique which is the term presence. In term presence, each feature is assigned a binary value which is set to 1 if that word appeared at least once in the article or 0 otherwise.

[4] discussed the use of a feature weighting method coupled with feature selection for text classification using a machine learning approach. TF-IDF was one of the feature extraction techniques used in their approach. Same as the work of [4], [5] also used TF-IDF as a feature extraction technique to create a general purpose polarity lexicon adaptive to any domain. In our work, we used TF-IDF to select words to be used as features in our dataset.

Some of the systems develop for subjectivity classification employing machine learning approach uses machine learning algorithms. Some of this algorithm includes Naïve Bayes [3][4][6][8][9], Maximum Entropy [3][6], Support Vector Machines (SVM) [3][4][6], Decision Trees [6] and Multilayer Perceptron[9]. Our work only focused on the Naïve Bayes, SVM, Decision

Trees and added k-Nearest Neighbor as machine learning algorithms.

III. METHODOLOGY

This section discusses the materials and methods used for this research undertaking. Our methodology involves 4 steps: (1) Data Collection, (2) Preprocessing and Feature Selection, (3) Feature Extraction and Model Building, and (4) Testing and Evaluation. Figure 1 shows the process of our work.

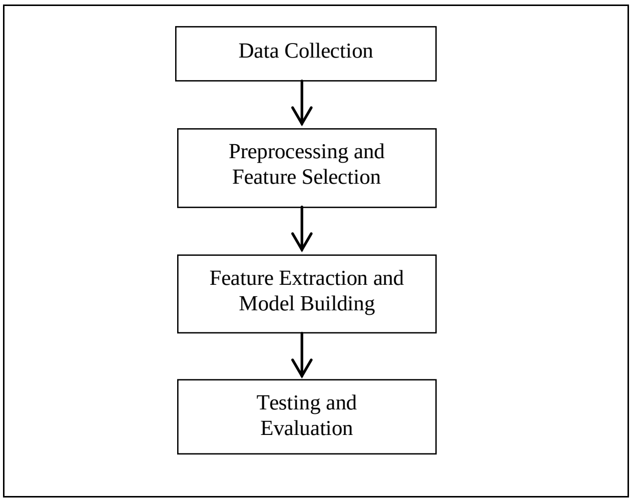
Data Collection
Preprocessing and Feature Selection
Feature Extraction and Model Building
Testing and Evaluation

Figure 1. Process flow

A. *Data Collection*

News and opinionated articles were collected to serve as the working corpus of this research. A web crawler was developed to mine articles from The Philippine Star's online portal. Articles labeled under the Bansa (news) and the Opinyon (editorial) categories and dated from 2001 to 2010 were collected. A total of 42,016 articles contributed by a total of 2,413 writers were collected during this process.

B. *Preprocessing and Feature Selection*

The features selected for model learning were words that received high TF-IDF scores in the opinion-based articles. TF-IDF is an algorithm that determines the relevance of a word in a set of documents or document queries [10]. It weighs the value of a word given a specific domain using the following formula:

$$w_d = f_{w,\,d} * \log(|D|/f_{w,\,D})$$

Where:
$f_{w,d}$ is the frequency of a word in a single document
$f_{w,D}$ is the number of documents that contains the word
D is the number of documents (1)

This algorithm weighs the relevance of a word not only based on the number of times it appears in a document. It also measures the relevance of a given word's existence in the whole set of documents related to a particular domain.

In order to get the TF-IDF scores the frequency count of each word in each article was counted. The news and opinion-based articles generated a total of 98,291 and 100,675 unique words, respectively. When combined together, the news and opinion articles had a total of 148,191 unique words. After counting the occurrence of each of these words in each article, the TF-IDF scores for the news and opinion domains were computed.

Words that received opinion-based TF-IDF score that was higher than 1000 as compared to their news TF-IDF score were chosen as features. A total of 162 words were used as features and these are listed in Table 1.

TABLE I. LIST OF WORDS USED AS FEATURES

Features
abangan, akin, aking, ako, akong, alagad, alam, ama, aming, anak, ano, asawa, atin, ating, ay, ayaw, ba, bagay, bahay, baka, bakit, balita, bang, basura, bata, bitag, bubuwit, buhay, cancer, com, dahil, daw, di, diyan, diyos, doktor, donya, eh, erap, este, gambling, ganito, ganitong, ganoon, ganyan, gaya, gma, gusto, he-he-he, hindi, ho, huwag, ibig, ikaw, inyo, inyong, itong, ito'y, iyan, iyon, iyong, jesus, jueteng, ka, kababayan, kahit, kaibigan, kami, kamote, kang, kanila, kanya, kapag, kasi, kaya, kayo, kayong, ko, kolum, kompanya, kong, kundi, kuwago, kuwagong, lahat, lalo, lang, lina, lupa, madam, maganda, mahihirap, mahirap, mali, man, mang, marahil, marami, maraming, matindi, may, meron, mismo, mo, mong, mrs, mukhang, nakausap, nakita, namin, nang, nangyari, nangyayari, natin, nating, nga, ngayon, ngayo'y, ngunit, nila, ninyo, ninyong, niya, niyang, noon, nung, nyo, o, ora, paano, pag-ibig, pala, palagay, paraan, parang, pati, pera, pero, po, porke, problema, pulis, raw, sabagay, sabi, sabihin, sagot, sana, sapagkat, sarili, senyora, shabu, siguro, sila, sila'y, sino, sir, siya, siya'y, siyempre, spo, subalit, suki, talaga, talagang, tama, tanong, tao, taumbayan, tayo, tayong, tingin, tiyak, totoo, tumawag, tungkol, wala, walang, yan, yata, yung

C. *Feature Extraction and Model Building*

For us to train the model, two datasets were created. One dataset to train the model for document level classification and one dataset to train the model for sentence level classification. Using term presence as the feature extraction technique each word feature of the instances in the dataset is assigned with a binary value which is set to 1 if that word appeared at least once in the article (for the document-level classification) / sentence (for the sentence-level classification) or 0 otherwise.

1) Document-Level Dataset

Since the goal of the model is to determine if a given document is objective or subjective. The dataset used for the document-level classification is comprised of articles from 2001 to 2010. News articles were labeled as objective while opinionated articles were labeled as subjective. The created dataset is balanced to ensure that they contain the equal number of objective and subjective articles.

2) Sentence-Level Dataset

For the sentence-level classification, the goal of the model here is to determine subjective sentences in an article. Opinionated articles from the year were split into sentences. To label the data, an expert was asked to determine if a sentence is either subjective or objective. Same as the previous dataset, this dataset was also balanced to ensure that they contain the equal number of objective and subjective sentences.

These two datasets were used to produce models through Weka [11] using different machine learning algorithms, which are as follows: C4.5 (decision tree), Naïve Bayes, k-Nearest Neighbor, and Support Vector Machine.

D. Testing and Evaluation

The models built were evaluated using 10-fold cross validation. 10-fold cross validation is done by splitting the data set into 10 near equal sized data sets. The system then iterates through 1 to 10, wherein each iteration, 1 data set is set aside and the rest is used for training. The data set that has been left aside is then used for testing. This is done until all data set has been used as the test set. The performance measure of each set up is saved and averaged to compute for the performance measure of the prediction model given the entire data set.

IV. Experiments

A. Document-Level Subjectivity Classification

Table II shows the result of a 10-fold cross validation experiment on the document-level subjectivity classification using C4.5, Naïve Bayes, k-NN, and support vector machines classifiers.

TABLE II. Results of Document-Level Subjectivity Classification

Algorithm	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	91.79%	0.918	0.918	0.918
KNN-1	83.84%	0.875	0.838	0.834
KNN-2	77.80%	0.844	0.778	0.767
KNN-3	76.64%	0.839	0.766	0.753
KNN-4	73.84%	0.826	0.738	0.720
KNN-5	72.89%	0.823	0.729	0.708
C4.5	92.65%	0.926	0.926	0.926
SVM	95.06%	0.951	0.951	0.951

Support vector machine has produced the best performance among the algorithms. The reason why support vector machine was able to perform well is because of the nature of the features. The word features were selected based on the words that are very associated with opinionated articles. Because these words are the words that separate most opinionated articles from news articles, the data set is most likely to be almost linearly separable. Support vector machine works by trying to find a line that divides the data into classifications, which is why the model performs very well for this particular set of features.

On the other hand, the k-nearest neighbor and the Naïve Bayes results are lower when compared against the other algorithms. The k-nearest neighbor performed poorly because the nature of the approach is to find the closest data that matches the unknown instance. This is not very appropriate in a data set which has a lot of features with 0 values. Naïve Bayes, on the other hand, has problems because the features are not independent of one another. Naive Bayes works under the assumption that the features are independent of one another. That is, the frequency of a certain word in the document does not affect the frequency of any other word in the document. This is not the case for written articles, are there are different likelihoods of certain words appearing with certain other words. This is one of the reasons why Naïve Bayes performs slightly worse than the other algorithms.

It is established that most of the results got a very high accuracy. This shows that in the task of classifying Filipino documents as subjective or objective, and the use of significant 'determining' words as features is effective.

B. Sentence-Level Subjectivity Classification

Table III shows the result of a 10-fold cross validation experiment on the sentence-level subjectivity classification using C4.5, Naïve Bayes, k-NN, and support vector machines classifiers.

TABLE III. Results of Sentence-Level Subjectivity Classification

Algorithm	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	58.75%	0.589	0.588	0.586
KNN-1	56.90%	0.583	0.569	0.550
KNN-2	56.40%	0.581	0.564	0.540
KNN-3	56.55%	0.583	0.566	0.542
KNN-4	56.20%	0.582	0.562	0.533
KNN-5	55.75%	0.577	0.558	0.528
C4.5	58.25%	0.589	0.583	0.574
SVM	58.30%	0.596	0.583	0.568

It is noticeable that all the results are lower, compared to the document-level classification. Thus we can conclude that the 'determining' words as features help determine if an article is subjective or objective, but not determine if a given sentence is subjective or objective. It was also noted that the list of words generated using TF-IDF contains words that are not only opinionated but also words that are not opinionated.

V. Discussions and Further Work

In this paper we have explored different machine learning algorithms on the task of classifying Filipino articles and sentences as subjective or objective. We used word features that appear more often with opinionated articles in building models using C4.5, Naïve Bayes, k-nearest neighbor, and support vector machine.

The different machine learning algorithms have all performed well in document-level subjectivity classification, showing that using certain determining words that can be found in the articles as features is effective in the task of classifying Filipino articles as subjective or objective. Among the models, we have seen that the support vector machine gives the best performance. Different machine learning algorithms have different advantages and disadvantages depending on the type of data that they work with, and the task that they are trying to accomplish. In this case, the fact that the data set is bound to be almost linearly separable allows the support vector machine to produce the best results among the different approaches.

On the other hand, applying the same set of determining words as features on sentence-level subjectivity classification did not produced the same results. A probable reason for this is that the words used are biased towards the document level classification since the TF-IDF process was performed on the document-level.

Despite having achieved the high accuracy with the produced models, this research field still has room for expansion. The researchers noted the following suggestions and areas of improvement for future research undertakings:

- Test the models on data from different domain
- Experiment on the TF-IDF threshold used to determine the words that will act as features
- Perform the TF-IDF on the sentence-level to validate the assumption above.

References

[1] B. Liu, “Sentiment Analysis and Subjectivity.” In N. Indurkhya, & F. J. Damerau, Handbook of Natural Language Processing, Second Edition. 2010.

[2] B. Pang, and L. Lee, “Opinion Mining and Sentiment Analysis.” Foundations and Trends in Information Retrieval, vol. 2, 2008, pp. 1-135.

[3] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: Sentiment Classification using Machine Learning Techniques.”, Proceedings of the ACL-02 conference on Empirical methods in Natural Language Processing, vol. 10, 2002, pp. 79-86.

[4] T. O’Keefe, and I. Koprinska, “Feature selection and weighting methods in sentiment analysis”. Proceedings of the Australasian Document Computing Symposium, 2009, pp. 67.

[5] G. Demiroz, B. Yanikoglu, D. Tapucu, and Y. Saygin, “Learning Domain-Specific Polarity Lexicon”. In proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops, 2012, pp. 674 – 679.

[6] S. Schrauwen, “Machine learning approaches to sentiment analysis using the dutch netlog corpus.” CLiPS Technical Report Series, Computational Linguistics & Psycholinguistics. 2010.

[7] M. Abdul-Mageed, S. Kübler, and M. Diab, "Samar: A system for subjectivity and sentiment analysis of arabic social media." In Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, 2012 ,pp. 19-28.

[8] S. Das, and M. Chen, "Yahoo! for Amazon: Sentiment extraction from small talk on the web." Management Science 53, no. 9, 2007, pp. 1375-1388.

[9] R. V. Regalado and C. Cheng, “Feature-Based Subjectivity Classification of Filipino Text”, Proceedings of the 2012 International Conference on Asian Language Procesing, 2012, pp. 57 – 60.

[10] K. Jones, “A Statistical Interpretation of Term Specificity and Its Application in Retrieval.”, Journal of Documentation, vol. 60, 2004, pp. 493-502

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update." ACM SIGKDD Explorations Newsletter 11, no. 1, 2009, pp. 10-18.