Negation-Based Sentiment Analysis in Reviews

Lam Dang
Department of Computer Science
Earlham College

[Introduction/Background]

Sentiment Analysis is the identification and determination of the emotion within the context of the texts. While researchers have been successful at making the machine identify the emotion based on positive/negative keywords (such as like, hate, love, detest, etc…), they are researching on an ongoing problem of how negation word (i.e not, don't, won't, didn't, hardly, etc...) will affect the emotion of the text. A negation word, consisted of many phrases, can either amplify or negate the emotion that was in the text. Therefore, knowing how to handle them correctly will increase the accuracy of sentiment analysis.

[Goals/Objectives/Hypothesis]

My work is on handling the negation to improve Sentiment Analysis in the context of Reviews. Most, if not all, reviews usually convey an emotions toward a subject. My plan is to identify the emotion and the negation word in the reviews, establish a relationship between the two words and based on the relationship, output an indication whether the review is positive or negative.

[Algorithm]

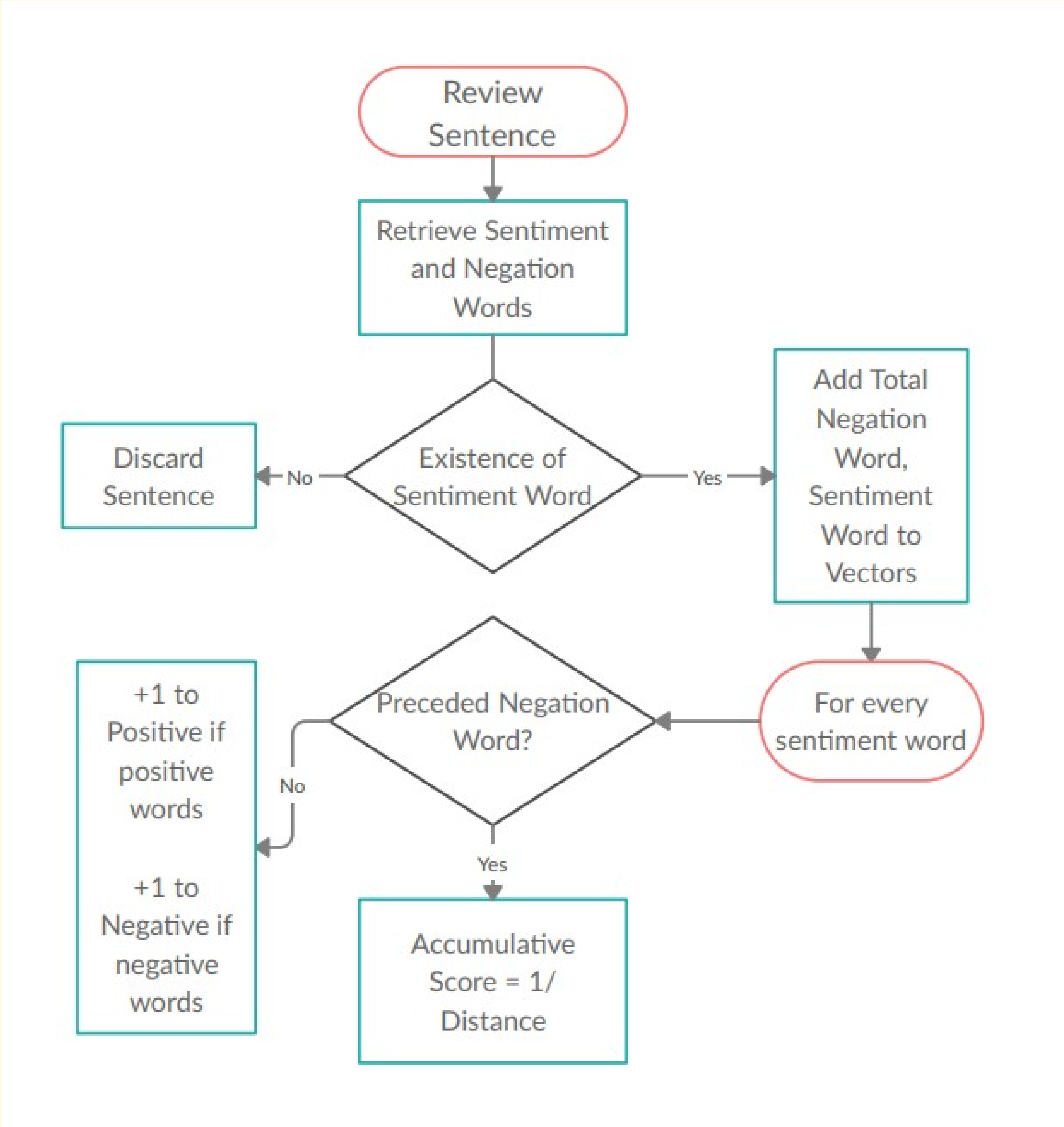

Review Sentence
Retrieve Sentiment and Negation Words
Existence of Sentiment Word
No
Discard Sentence
Yes
Add Total Negation Word, Sentiment Word to Vectors
For every sentiment word
Preceded Negation Word?
No
+1 to Positive if positive words
+1 to Negative if negative words
Yes
Accumulative Score = 1/ Distance

Figure 1: Flow Chart of The Preprocessing Algorithm

[Methods]

Stage 1: Data Collection:

The Data Set used to train and to validate the model is the Amazon User Review. The Amazon User Review contains millions of text reviews, divided into multiple categories. The model for this thesis will be trained and validated with each categories of the data set.

Stage 2: Data Preprocessing:

The five-stars score system will be divided into three labels: “Bad”, “Neutral”, “Good”, which depict the customers’ satisfactions of a product.

The text review will be processed as follow: For every sentence in one review, scan for negation word and sentiment word, keeping track of the in respectable arrays. If a sentence does not have any sentiment word, discard the sentence. If sentiment words exist, determine whether a negation word preceded it. Calculate the positive/negative score based on the distance between the negation word and sentiment word. The scores will be a key value to determine an user’s satisfaction to a project.

Stage 3: Model Training:

Logistic Regression use the Sigmoid function to calculate the possibility of a data point belongs to one class or not. All the processed data points from our dataset will pass through the model, making the model reconfigure the Sigmoid function that corresponds with the class and the datapoint features.

Neural Network consists of multiple layers and nodes that are tasked to identify similarity between datapoint and classify them into a same class, if the similarity in a certain feature is found. When you pass the data to the neural network to train, it will identify from a range of data point with the same class, which feature they have are similar, making those feature a key identifier for the class.

Stage 4: Model validation:

Using a different set of data point and class, we can pass these data point through the trained model, let it output a predicted class and compare it with the actual class given by the customer.

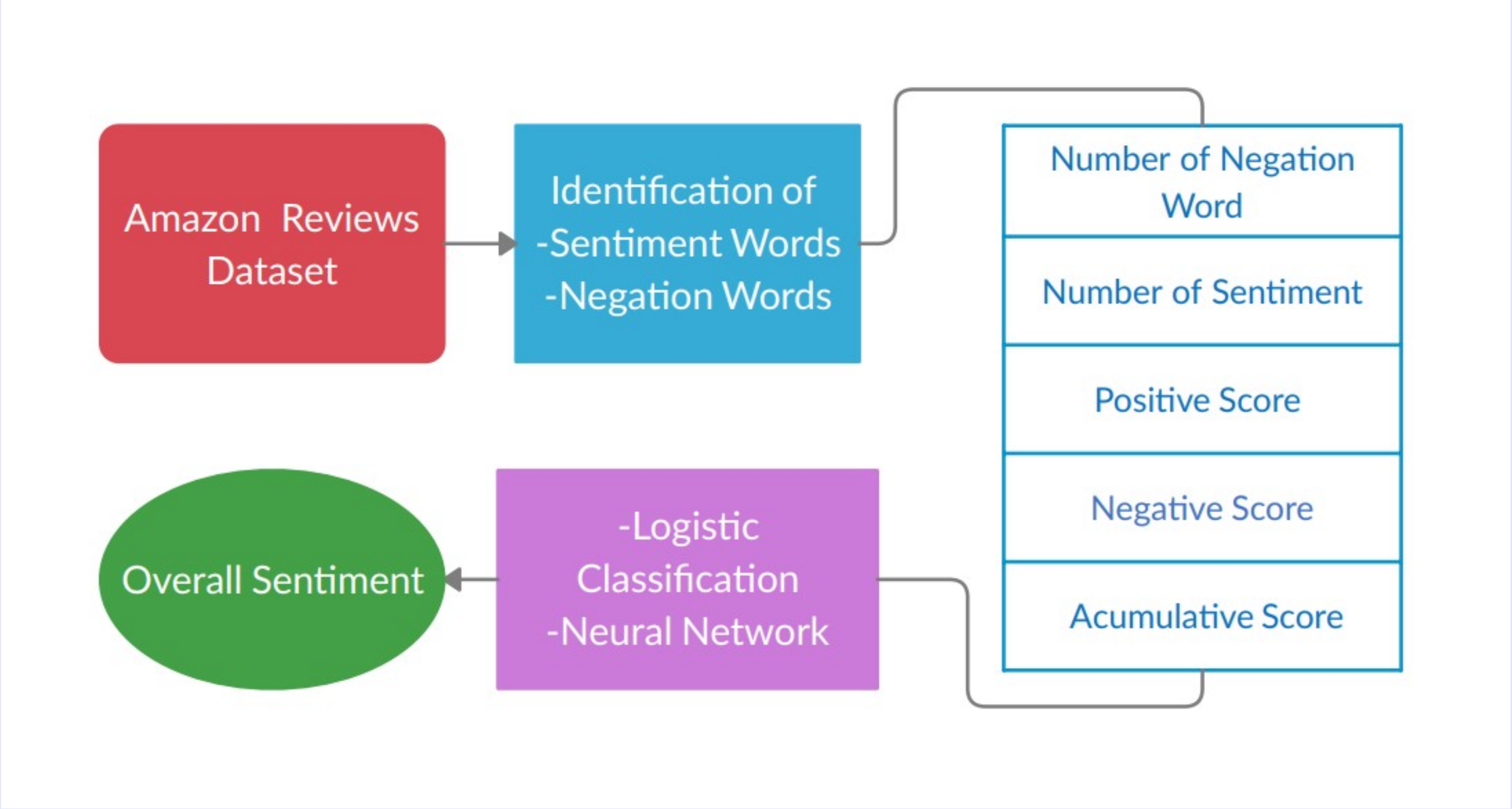
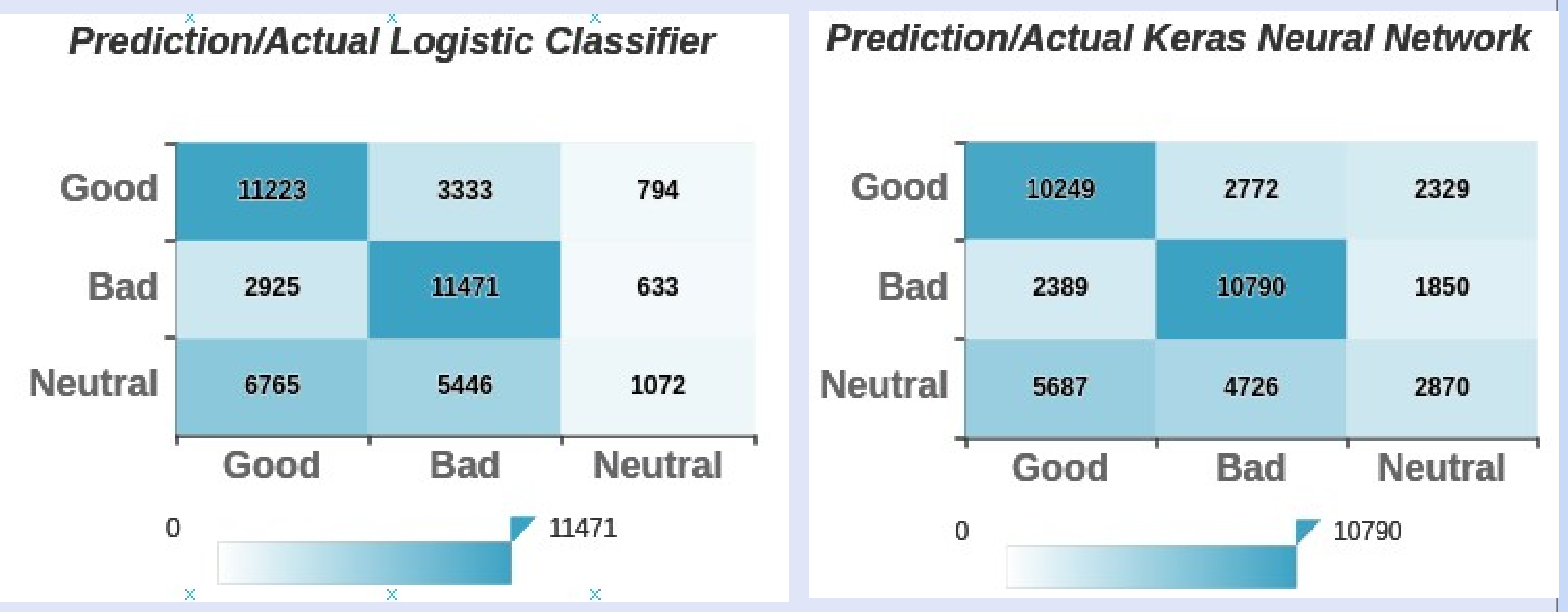


Figure 2: Framework of the model

[Result]

- Predictions of Reviews labeled as Good (4-5 stars) and Bad (1-2 stars) performed well, achieving 70-80% accuracy scores
- Predictions of Reviews labeled as Neutral (3 star) performed poorly, achieving less than 20% accuracy.

The poor performance of the Neutral Predictions was caused by the multiple classifications of the class. As the Neutral class is on the border of Good and Bad classes, the ideal scenario, and the scenario that my algorithm is aiming for, is that the amount of negative and positive indications are balance, leading to a “Neutral” predictions. However, as the result shows, only less than 20% of the case are so, and categorized as a Neutral class. In many cases, even though the positive and negative indications were not balanced, the customers would still give it a Neutral emotion class.



Prediction/Actual Logistic Classifier

	Good	Bad	Neutral
Good	11223	3333	794
Bad	2925	11471	633
Neutral	6765	5446	1072

Prediction/Actual Keras Neural Network

	Good	Bad	Neutral
Good	10249	2772	2329
Bad	2389	10790	1850
Neutral	5687	4726	2870

Figure 3: Heat map of True/False predictions of the two models

[Future Work]

- Study the pattern of "Neutral" class and add it as a feature such that when the pattern appeared, the data point will be categorized correctly, thus improving the accuracy of the model

- Use multiprocessing to parallel the process of preprocess data, which will overall enhance the speed and performance of the algorithm.

[Acknowledgements]

I would like to thank Professor Igor Minevich and Professor David Barbeda for their expertise on the subject of Natural Language Processing and Mathematics.

I also would like to thank Professor Xunfei Jiang and Professor Charlie Peck for their advisement on research works, topic suggestions and paper/code/poster reviews.