ASPECT BASED SENTIMENT ANALYSIS FOR PERSONAL MARKETING

July 19, 2019

Armando Banuelos
Stanford University

Introduction

The purpose of this report is to go over a detailed summary of the Aspect Based Sentiment Analysis (ABSA) model built for the purpose of personal marketing of automobile services. This report will be structured in a way for lay-people and the technically-inclined to both understand the nuances of the architecture composing the model. If questions should arise, please feel to contact me at armandobn9@gmail.com.

Finding the Need

When automobile mechanics are servicing a car, they have to write down a summary of services they performed on the car along with their recommended services and other observations. This compiled information is then fed to the marketing committee of the car dealership where they read over these notes to see what deals/offers they can send to their customers related to the recommended, completed, or declined services mentioned in the report.

The purpose of the ABSA model is to cut out the middle man and to instantly send marketing campaigns once the automobile mechanic completes their report using the power of Natural Language Processing (NLP). NLP is a subfield of computer science and artificial intelligence concerned with the processing and analysis of large amounts of written or spoken language. Through the power of artificial intelligence, car dealerships can optimize the way they market their services to their customers.

What is Aspect Based Sentiment Analysis?

Before we can introduce the concept of aspect based sentiment analysis, we must first understand sentiment analysis. Sentiment analysis is the process of predicting whether the polarity of a given sentence or series of sentences is positive or negative. For example, consider the sentence: "I hate math". NLP algorithms that implement sentiment analysis can be constructed to correctly predict that there exists a negative polarity in this example. This process is referred to as sentiment analysis.

Aspect based sentiment analysis applies that same logic at a more granular level. ABSA is concerned with finding the polarity of a given aspect or word in a sentence. For instance, in automobile mechanic reports you may see: "The cabin air filter is fine but the rear brake pads need replacement." An NLP algorithm that applies ABSA correctly will show that the aspect "cabin air filter" has a positive polarity whereas "rear brake pads" has a negative polarity.

The ABSA model presented in this paper aims to classify the sentiment of different aspects in automobile mechanic reports. The aspects the model includes many of the typical services provided by car dealerships along with most popular car parts. Examples include tires, brakes, transmission fluid, oil change, and so forth.

So how exactly does the model work?

To preface this section, the way this model works is very complicated. For those wanting to dive deeper into the model, please refer to the GitHub Repository and navigate to Techical Report.pdf.

Here, we will go over a more intuitive and less technically intensive understanding of what the model is doing under the hood. For the creation of current ABSA model, there were two models used as points of comparison. The first was n-grams bag of words ABSA and the second was content attention based aspect sentiment analysis classification (CABASC).

N-grams Bag of Words ABSA

The way the n-grams bag of words ABSA model works is that it analyzes the sentiment of a particular aspect or word by looking at the sentiment of the n words to the right and n words to the left of the current aspect of interest. For instance, consider the report: "Cracked window but the windshield wipers are still intact." If the aspect of interest is "windshield wipers" and if $n = 4$, we notice that the n words left of our aspect expresses a negative polarity through the use of words like "cracked" and "but" whereas the n words right of our aspect expresses a positive polarity using words like "intact". To determine whether our aspect "windshield wipers" has a negative or positive polarity, a weighted average has to be constructed with the words "intact", "but", and "cracked". In this case since more negatively polarized words exist, n-grams bag of words will report an overall negative sentiment analysis for windshield wipers. However, this is incorrect since windshield wipers are fine; it is the window that needs to be fixed not the wipers themselves.

As you can probably see, there are limitations to the use of the n-grams bag of words model. Overall accuracy of this model on the given dataset was 34%. The first limitation is that you can only see at most $2n$ words to formulate your hypothesis of the polarity of the aspect. At times you may need to see more than just $2n$ words to determine polarity, especially given that automobile mechanic reports can be extensive and detailed. Secondly, the weighted average of polarized words within the $2n$ sequence of words encapsulating your aspect may, more often than not, incorrectly predict the polarity. The use of highly polarized words like "no", "not", or "never" may skew the predicted result to incorrectly predict polarity (ie "No need to fix the brakes").

Content Attention Based Aspect Based Sentiment Analysis (CABASC)

The CABASC model was cleverly created by Qiao Liu in 2018. This model makes use of neural networks. Neural networks in the realm of NLP are ways to keep track of information from multiple points of views of a sentence. In the n-grams bag of words model, we were only able to analyze $2n$ words at a time. Neural network architectures allow us to keep track of all parts of a sentence. The way neural networks do this is by converting a series of words to numerical matrix equivalents referred to as a word embedding. Linear algebra is applied to these matrices to discover patterns and to ultimately learn about the relationship between different parts of the matrix, hence different parts of the sentence. The way CABASC works is that it passes in the automobile mechanics report and a list of all the aspects located in the sentence, computes some math on the word embeddings constructed to create a probabilistic prediction of polarity for all aspects. For more information, please see the Github Repository. Overall accuracy of this model is 60%.

Best Model: Attention Based Long Term Short Term Memory ABSA (ATAE-LSTM)

The final model that generated the best testing accuracy was the attention based long term short term memory. Similar to the CABASC model, the sentences from the report of the model are translated to numerical equivalents wherein linear algebra is applied to recognize patterns within the matrices. The reason why this mode outperforms the the CABASC model is that there exists a layer within the neural network known as a long short term memory (LSTM) memory, which has the capacity to recognize patterns between words regardless of the distance between them. For instance, a sentence might refer to the polarity of a service near the end but mention said service near the beginning. An example includes: "The air conditioner seems to be in a state of constant deactivation which can cause iteration for client since they are heavily allergenic to pollen. Recommended action to approve. Customer has declined." In this example we see that the LSTM layer is able to retain information that the recommendation and the customer declining pertains to the air conditioner aspect regardless of the words between them. This is particularly helpful especially when accurately reporting the polarity of aspects within sentences. Overall accuracy achieved is around 80%.

Fine Tuning the ATAE-LSTM Model

Although the model reports an 80% accuracy, this is an average across all services. However some services like oil change have very consistent aspect patterns such as "suggest oil change" or "oil fluid leaking" and as result, reports more accurate results. The last step of the model is about ensuring that this fine tuning is applied. For that reason, a model was created for each servicing category in order to address this and to showcase the efficiency of the model.

How to run the demo?

For more information on how to train and test the model on your own data, please refer to the GitHub Wikipedia. Within there should be more information on how to handle that information.