Aspect Based Sentiment Analysis for Personal Marketing Solutions

Armando Banuelos
GitHub Repository Here
Stanford University
abanuelo@stanford.edu

Abstract

Sentiment analysis has traditionally been used to predict the positive or negative polarity of a given sentence(s). A more complicated task is to predict the aspects mentioned in a sentence and the sentiments associated for each of them. This task is known as aspect-based sentiment analysis (ABSA). The goal of this paper is to integrate and improve recent neural architectures to identify the sentiments of different aspects of an automobile service report in order to create personalized marketing solutions.

1 Introduction

The goal of this paper to develop and evaluate a real-time personal marketing system for the automobile industry using aspect based sentiment analysis (ABSA). Given a historical report of the auto services performed on a vehicle, this system attempts to identify and label each service as completed, declined, neutral, or recommended. In this paper, a service will be referred to as an aspect and the subsequent parts, actions and/or adjectives will be the features for said aspect. For instance, an oil and filter service is the aspect and words 'leak', 'oil', 'filter', etc are the features. For those services labeled as declined or recommended, personalized marketing can be delivered to the user to remind them to take action or make appointments to do them in the near future. The historical service report data includes 949,700 individual reports courtesy of Porsche. More elaboration of the data will be provided in the data section.

2 Related Work

Among the officially published work, the task of aspect based sentiment analysis has evolved from supervised models to deep learning methods. Zhang, Xu, and Wan developed a preprocessing and candidate feature technique that could be used to more easily determine the sentiment of an aspect within a sentence [1]. Similarly Wang and Liu integrated a deep learning method that integrated a 2-layer convolutional neural network (CNN) along with a parse tree that served to learn the sentiment of h consecutive words to be used to evaluate the sentiment of an aspect in a given sentence [2]. Their aspect based sentiment architecture achieved an F1 Measure of 0.513. More recently, Liu et. al. developed a content attention model for ABSA that achieved an accuracy of 0.8 [3]. Wang et. al. implemented an LSTM layer with attention to generate promising results of 0.72 for a three way class prediction [4].

Inspiration to expand upon the work of Liu et. al. and Wang et. al. stems from the their highly effective results in ABSA tasks. More specifically, interest in using an LSTM layer in the architecture comes from their ability to learn long-term dependencies and contextual features from previous and future states which can give a more holistic sentiment evaluation for a given aspect. As a baseline, Zhang's et. al. work will be incorporated to extract relevant features for the subsequent creation a simple bag-of-words model for the ABSA task.

3 Approach

3.1 Baseline: Ngrams Bag Of Words

Let $s = \{s_1, s_2, s_3, ...s_n\}$ consist of the different services an automobile servicing agency performs such as oil and filter change, brake change, and general multi-point inspection just to name a few. Every service s_i will have corresponding aspects $a = \{a_1, a_2, a_3, ...a_n\}$. For instance, the service for brake change can have the aspects brake pad, thread, and brake oil among many others. Each s_i can be mapped to a sentiment space f = {-1,0,1,2} where $f = -1$ corresponds to a declined service, $f = 0$ corresponds to a recommended service, $f = 1$ corresponds to a completed service, $f = 2$ corresponds to a neutral service. Each f_i has a feature list $x = \{x_1, x_2, x_3, ...x_n\}$ that increases the probability that a specific a_i has sentiment f_i. Let $r = \{r_1, r_2, r_3, ...r_n\}$ correspond to the set of reports for an individual vehicle. For a sequence of h ngrams in each r_i, a list of aspects a and a list of features x will be generated and mapped to s to determine sentiment f for service type.

3.2 Attention-Based LSTM with Aspect Embedding (ATAE-LSTM)

For each automobile servicing report s_i in $s = \{s_1, s_2, s_3, ...s_n\}$, aspects $a = \{a_1, a_2, a_3, ...a_n\}$ will be generated. Each s_i will form a word embedding $w_i = \{w_1, w_2, w_3, ...w_n\}$ of dimension d and aspects a will form an aspect embedding $v_i = \{v_1, v_2, v_3, ...v_n\}$ of dimensions d_a. Word embedding w_i and aspect embedding v_i will be concatenated as input into the LSTM cell layer. More formally, each cell in the LSTM can be calculated as follows:

$$X = [h_{t-1}, x_t]^{-1} \tag{1}$$

$$f_t = \sigma(W_f \cdot X + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot X + b_i) \tag{3}$$

$$o_t = \sigma(W_o \cdot X + b_o) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c \cdot X + b_c) \tag{5}$$

$$h_t = o_t \odot tanh(c_t) \tag{6}$$

where $W_i, W_f, W_o \in \mathbb{R}^{dx2d}$ are the weighted matrices, $b_i, b_f, b_o \in \mathbb{R}^d$ the the biases, σ is the sigmoid function, $\odot$ is elementwise multiplication, x_t is the input to the LSTM Cell unit, and the vector of the hidden layer is h_t. In the case of the ATAE-LSTM, the W_f can be decomposed as a representation of w_i and v_i. For word embeddings, a matrix $H \in \mathbb{R}^{dxn}$ will correspond to hidden vectors $\{h_1, h_2, h_2, ..h_n\}$. For aspect embeddings, each individual aspect v_i within the embedding will be multiplied to a weighted matrix W_v. Adding the attention mechanism will produce an attention weight vector α and weighted hidden representation r as follows:

$$M = tanh([W_h \cdot H, W_v \cdot v_i \otimes e_N]^{-1}) \tag{7}$$

$$\alpha = softmax(w^T M) \tag{8}$$

$$r = H\alpha^T \tag{9}$$

where $M \in \mathbb{R}^{(d+d_a)xn}$, $\alpha \in \mathbb{R}^n$, $r \in {}^n$ and $W_h \in \mathbb{R}^{dxd}$, $W_v \in \mathbb{R}^{d_a x d_a}$, $w \in \mathbb{R}^{(d+d_a)}$ are projection parameters, and e_N is column vector 1's of dimension N. The last step is the retrieve the final sentence representation h^*, pass it through softmax layer along with linear layer to find the conditional probability distribution y of sentiments as follows:

$$h^* = tanh(W_p r + W_x h_N) \tag{10}$$

$$y = softmax(W_s h^* + b_s) \tag{11}$$

where W_p and W_x are parameter learned during training. The image of the architecture is below.

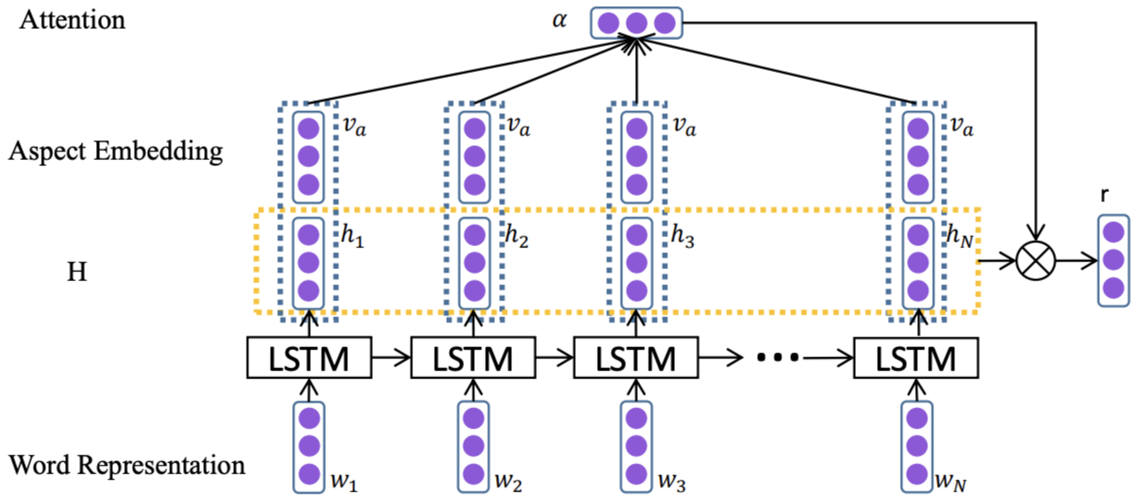

Figure 1: Architecture for the ATAE-LSTM. Read section above for more details.

3.3 Content Attention Based Aspect Based Sentiment Classification (CABASC)

Given a service report s_i of dimension d, we will divide it into two: left of the aspect a_i, s_{ileft} = $\{z_1, z_2, z_3, ...z_l\}$ and right of the aspect a_i, s_{iright} = $\{z_r, z_{r+1}, z_{r+2}, ...z_n\}$ where z_i represents an individual word within s_i. These two sections will have their own word embedding representations labeled E_{left} and E_{right}. Now, to learn how contextual words and aspect relate to one another, we will pass each embedding into a GRU cell layer as follows:

$$r_t = \sigma(W_r e_t + U_r h_{t-1}) \tag{12}$$

$$z_t = \sigma(W_z e_t + U_z h_{t-1}) \tag{13}$$

$$\hat{h_t} = tanh(W_h e_t + U_h(r_t \odot h_{t-1})) \tag{14}$$

where $W_r \in \mathbb{R}^{dxd}$, $W_z \in \mathbb{R}^{dxd}$, $W_h \in \mathbb{R}^{dxd}$, $U_r \in \mathbb{R}^{dxd}$, $U_z \in \mathbb{R}^{dxd}$, $U_h \in \mathbb{R}^{dxd}$ are weight matrices, e_t is an individual element within E_{left} or E_{right}, r regulates capacity to update new hidden state, $\hat{h_t}$, and z_t controls how much information from previous state h_{t-1} is prohibited. GRU cell states to both E_{left} and E_{right} generate hidden states H_{left} and H_{right}. An MLP layer is applied to each H_{left} and H_{right} as follows:

$$\beta_{lr} = \sigma(W_d h_{lr} + b_d) + b_{lr} \tag{15}$$

where β here is generalized for both $\beta left$ and β_{right}, $W_d \in \mathbb{R}^{1xd}$ is a generalized weight matrix, $b_d \in \mathbb{R}$ is a generalized bias, b_{lr} is the basic attention bias for both the left and right hidden states, and h_{lr} is an individual hidden states within either H_{left} and H_{right}. Now that $\beta left$ and β_{right} have been extracted the next step is to concatenate those weight vectors and multiply them by a scalar of 0.5 as follows:

$$\beta = (\beta_{left} + \beta_{right}) * 0.5 \tag{16}$$

As final step to the architecture, we take compute memory $M = \{m_{w1}, m_{w2}, m_{w3}, ...m_{wn}\}$ to gather the conditional probability of sentiments for aspect a_i:

$$m_{wi} = \beta_i \odot m_i \tag{17}$$

The image of the subsequent model is provided below.

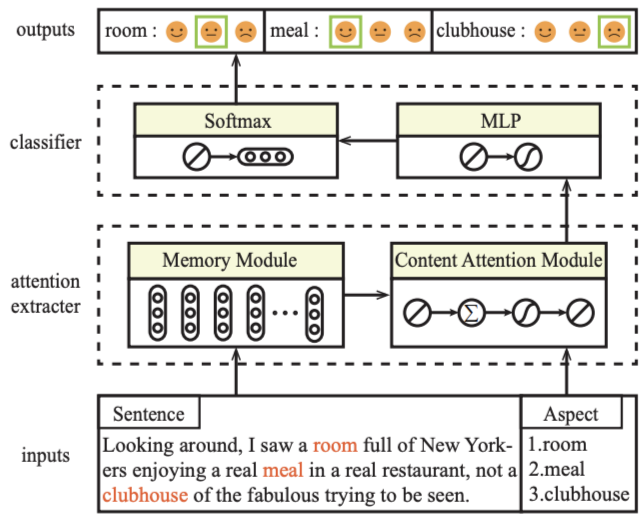

Figure 2: Architecture for CABASC. Read section above for more details.

4 Experiments

4.1 Dataset and Data Preprocessing

As mentioned within the introduction of this paper, the dataset includes 949,700 individual vehicle reports courtesy of Porsche. An example of an individual data point is:

"ID: 7856334; Comment: Coolant has leak causing ac malfunction | Performed multi-point inspection 7mm left brake pad, 6mm right brake pad, tire thread wear, recommend replacing | replaced left and right tires"

The data has a corresponding ID along with a comment where each individual comment can be delimited via a pipe. For data preprocessing, table of service type along with aspects associated with service type needed to be created. The aspect-feature table was generated and can be viewed at the appendix of this report. Along with that, data had mention of highly technical numbers such as psi, mm, and mileage quantities that were irrelevant for the aspect based sentiment analysis. Data was pre-processed such that to keep soley alphabetic characters.

4.2 Evaluation Method

Since there are so many service types and aspects, the best form of evaluation method will be used accuracy with n-feature space. I will measure all those correct labels to the data. Data will be trained and then evaluated on a total of 60,000 examples. The section titled Results will hold more of that information.

4.3 Experimental Details

Data was split into 80% train and 20% test. More so, the data is labeled into different sentiments. Those sentiments are performed, declined, and recommended. The distribution between these three polarities within the training data was around 65% percent perform, 20% recommend, 10% decline, and 5% neutral. The unequal distribution of data polarities were adjusted by a greater penalizing loss for classes with more data.

The ATAE-LSTM and CABASC models had learning rate of 0.001, Adam optimizer as the optimizer, embedding dimensions of 300, hidden dimensions of 300, polarity dimensions of four. For now, all models were run locally for a total of 2000 epochs.

4.4 Results

As indicated by the table below, the results showed that the ATAE-LSTM model produced the overall best accuracy on test data compared to the baseline ngrams bag-of-words-model along with the the CABASC model. Reason as to why the LSTM performed best was its ability to hold long-term dependencies between the individual reports between different aspects. More so, the right have learning rate for the ATAE-LSTM allowed for faster convergence, which contributed to the overall success of the model.

Model	Accuracy
Ngrams Bag-of-Words	0.34
ATAE-LSTM	0.8
CABASC	0.6

4.5 Analysis

When analyzing the overall success of both models, we see that there seems to be more inaccuracy when it comes to analyzing longer report narratives. Reason for this holds implicitly from the LSTM and GRU architectures which enable features to from the longer reports to be 'remembered' by the model. For instance consider the example:

"| 45051 per customers request tires are unsafe to drive customer declined tires at time 900237.| 45051 found p0421 and p0431 cat efficiency below threshold both banks 600 scanned vehicle found p0421 and p0431 cat efficiency below threshold both banks, cleared faults and performed hort test and found failed again, checked and found after market cat converters in both banks, removed both cats and replaced with new ones and hardware, cleared faults performed short test found all passed ok, no faults at time 900237.| 45051 required fronts 29 rears 31 psi torqued wheels to manufacture specs tires are unsafe to drive customer declined repairs 900237.| 45051 per repair order performed vehicle inspection found windshield cracked, 4 tires need replacement fronts at 4mm rears less than 1mm unsafe to drive, 4 wheel alignment, 2 year brake fluid service due, tb injection service due, major service due, serp belt due to time, customer declined all repairs at time 900237".

When feeding in the overall report after it has prepossessed to remove nonalphabetic characters, we get that the sentiment of certain aspects such as Belt Service get marked as completed when it should be marked as declined. Reason to believe that the LSTM captures positive sentiment from these feature based on other reports. For instance, mention of performed from the second piped report along with mention of "performed vehicle" out-weight the presence of negative features like "decline".

Another thing when analyzing the performance of the model is that is handles double negatives very poorly. For instance when inserting mention of "The declined was not gone through", the model will report that as declined when it should not. Reason to believe that there is a lack of training data that holds the double negative property.

5 Conclusion

From the three models tested, the ATAE-LSTM model performs best with total of 0.8 accuracy generated on testing data. Both CABASC and ATAE-LSTM models perform poorly due to the architectures tendency to hold long-term dependencies between word features. When more instances of performed behavior appear within the long reports, a current aspect's polarity gets overshadowed by those results. Both models perform poorly on double negatives since there is a lack of training on double negatives.

For future work, recommended action to create an ATAE-LSTM and CABASC model for each service type to fine grain details. For instance, the category of Multipoint Inspection will get trained by with its own data and and tested on its own data using the same architecture mentioned in this paper. There exists reason to believe that by doing so, model can be more fine-tuned to 'pay-attention' to the features of its own data and not of the data of other servicing categories as it currently stands.

6 References

[1] Zhang, Wenhao, Hua Xu, and Wei Wan. "Weakness Finder: Find product weakness from Chinese reviews by using aspects based sentiment analysis." Expert Systems with Applications 39.11 (2012): 10283-10291.

[2] Wang, Bo, and Min Liu. "Deep learning for aspect-based sentiment analysis." Stanford University report (2015).

[3] Liu, Qiao, et al. "Content attention model for aspect based sentiment analysis." Proceedings of the 2018 World Wide Web Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2018.

[4] Wang, Yequan, Minlie Huang, and Li Zhao. "Attention-based LSTM for aspect-level sentiment classification." Proceedings of the 2016 conference on empirical methods in natural language processing. 2016.