Aspect Based Sentiment Analysis on Quickbooks Mobile AppStore reviews

W266: Natural Language Processing
Rishikesh Majumder and Harith Elrufaie
{rmajumder, harith.elrufaie}@berkeley.edu

Abstract

Today, product reviews are foundational in the success of any product. They are not only relevant to the prospective consumers who are in the pursuit of an unbiased opinion, but they are as important to the makers of the product. These reviews play an essential role in measuring the overall user satisfaction and also shed light on areas that require attention.

While today some of the online platforms provide sentiment polarity on the product reviews, it remains a challenge to understand the sentiment polarity on specific aspects of the product. In this paper, we developed a model that provides aspect sentiment polarity on Quickbooks Mobile reviews data. Using Capsule Network, the model utilizes a learning framework to transfer knowledge from the document-level to the aspect-level. Our trained model achieved an F1 score of 0.84.

1. Introduction

In recent years, sentiment analysis of product reviews became an emerging active area of research. Several papers proposed diverse approaches to determine the sentiment polarity (positive, negative or neutral) on product reviews [1, 2, 3]. However, what also became evident is that sentiment analysis at the document (review) level is no longer sufficient to gain precise insights on a specific product aspect. The opinion expressed on the topic is given significance rather than the topic itself [4]. In contrast, by performing sentiment analysis at the aspect-level, we can gain more insights into the various aspects of the product [5].

The target of opinion is referred to as an entity. An entity can have a set of aspects. For example, “iPhone” is an entity that has a set of aspects such as battery and screen. In the field of sentiment analysis, aspects are often referred to as features, product features or opinion targets [6].

1.1. Related Work

The concept of aspect-level sentiment analysis and opinion mining was first introduced in the year 2003 [7, 8]. These studies applied several strategies such as frequency-based, syntax-based, supervised and unsupervised machine learning to build a model that can classify sentiment at the aspect level.

The earliest work on aspect detection from online reviews was presented by Hu and Liu [9]. They provided a frequency-based approach that used association rule mining using part-of-speech (POS) to extract frequent noun phrases as product aspects. They used WordNet to identify opinion words. Similarly, Ana-Maria Popescu and Oren Etzioni [11] introduced an unsupervised information extraction system called OPINE which extracted all noun phrases from product reviews and kept those with a frequency greater than a defined threshold.

Alqaryouti et al [10] proposed a model that integrates several lexicons and rule-based techniques to extract users’ sentiments.

Schouten et al [12] on the other hand presented unsupervised and supervised methods to extract aspects based on co-occurrence frequencies. They used spreading activation on a graph built from word co-occurrence frequencies in order to detect categories.

Finally, a comprehensive survey of various aspect-level sentiment analysis approaches was published by Schouten and Frasincar [13]. The study highlights a number of influential approaches, evaluates the characteristics of proposed algorithms and compares the performance of each.

2. Project Overview

QuickBooks is an accounting software developed by Intuit [14]. There are nearly 5 million Quickbooks users around the world who use the product to run their businesses.

Examples of such activities are creating invoices, sales receipts, payment processing, paying vendors, running reports and tracking the hours of their employees.

With the number of reviews that are received on a daily basis, Product managers and teams spend a substantial amount of hours to parse, organize, categorize and analyze these reviews in order to take further actions. The makers of the software need a system that is capable of 1. Extract the aspects from the product reviews. 2. Predict the polarity on every identified aspect.

2.1 Dataset used

The review data are for the Intuit Quickbooks Mobile application. We have obtained approximately 20,000 review records from the iOS App Store and Google Play stores. The average number of words in the review is 142. Each review has an accompanying rating score that is given by the user. Also, each review has a document-level sentiment implying the overall satisfaction level.

2.2 Background and problem approach

In any review, aspects could be mentioned explicitly or implicitly. Consider this example, "Easy to use. I love how I can easily customize invoices". In this example, *the invoice* aspect is mentioned explicitly. In contrast, "Could be made a little easier to use." We can infer that the aspect is the "Experience." Therefore, our solution should be able to identify aspects whether they're mentioned or not.

One of the biggest challenges in this problem is the lack of available labeled data that can help us identify the right aspects for the product. We needed a dataset with labeled aspects and their sentiments in order to build and train a model that can make the polarity classification at the aspect level. As a result of this obstacle, we manually labeled a quantifiable amount of data so we can build a model that can identify the product aspects in a supervised fashion. We will explain the approach in further detail in the next section.

3. Problem Approach

Our approach is inspired by the Transfer Capsule Network (TransCap) work by Chen and Qian [15]. In their model, they use Inductive Transfer Learning (ITL) to transfer sentiment knowledge from Document Semantic Classification (DSC) to Aspect Semantic Classification (ASC).

Along with the contextual features and semantic constructs, positional embedding of tokens in each input is significantly effective in measuring sentiment polarity. We applied Gu et al. [16] approach in assigning weights according to the position of the word and how polarized the word is. With this modification, we achieved better scores (1-2%) in terms of accuracy and f1.

In the next section, we will review some of the key concepts behind our proposed model generation, then we will illustrate the proposed architecture.

3.1 Capsule Network and Dynamic Routing

Capsule Networks, proposed by Sabour et al. [17], consists of a group of neurons (vectors), called a capsule, which is added to a conventional neural network. However, unlike CNN, capsule network convolutional layers produce vector-output capsules and use routing by agreement (dynamic or adaptive routing) instead of detecting features with filter and pooling. A capsule network uses hierarchical relationships to choose the amount of information to pass to the next layer to maximize the activations of upper-level capsules. While passing feature information to upper capsule layers, it does not entirely rely on backpropagation, rather it runs through multiple iterations to aggregate multiple feature parameters independently and passes on to the next layer.

NLP classification tasks, such as aspect-based sentiment classification, can be effectively applied to use the transfer learning attribute of a Capsule Network, and to identify segments and associated aspects in a block of text without stressing on the entirety.

3.2 Model generation and two-step training process

3.2.1 Data Cleaning

Reviews data, in general, are not clean, therefore it is necessary to take the entire dataset through a data cleansing process. Our cleaning process included the removal of HTML tags and non-English reviews, lemmatization (using NLTK WordNetLemmatizer), removal of stop words and autocorrection of misspellings.

3.3.2 Aspect Extraction and Generation

As previously mentioned, one of the biggest obstacles is the lack of labeled data to train and build that model that can identify aspects of a product. Knowing the Quickbook product, we can easily name aspects such as invoicing, expenses, reports, and payments. However, after reading hundreds of reviews, it came into our attention that aspects should also include non-functional ones, such as quality, performance, and experience, as these aspects were implicitly mentioned in several reviews. Therefore our goal became: 1. Identify the functional and non-functional aspects that we want to capture. 2. Manually pre-label aspects on a number of product reviews to build and train the model.

For the first problem, we decided on the following aspects: App, Experience, Quality, Transactions, Updates, Receipt Capture, Invoicing, Estimates, Customers Service, Dashboard, Expenses, Subscription, Help, MileageTracking, Performance, Reports. For labeling aspects, we manually labeled 600 reviews, then used Spacy Named Entity Recognizer (NER) to update Spacy pre-trained model. In this process, we ensured that the examples are hydrogenous enough to avoid overfitting. See Fig 1 for sample model training data.

```
train_data = [
   ("I love the simple features on the app", [(11, 17, "Experience"), (34, 37,
   "App")]),
   ("Too often getting experiencing technical difficulties when trying to login",
   [(31, 51, 'Quality')]),
   ("Since the update I am not able to find the mileage tracking feature on the
   Android app", [(10, 15, "Update"), (43, 59, "Mileage Tracking")]),
   ("Makes invoicing on the go fast and easy. Just write it up and send it to their
   email", [(6, 15, "Invoicing")])]
```

Figure 1

We used the generated model to generate aspects for the rest of the reviews. Using Spacy named entity browser, we were able to fine-tune the generated model incrementally. Next figure displays some of our testing results. The highlighted text depicts the detected entity, whereas the text in bold and caps is the name of the classified entity generated by the model.

#1:

Great app **APP** for small business owners. I wish trial free trial **SUBSCRIPTION** was more than one month. The expenses **EXPENSES** are my favorite.

#2

Not very happy with the last update **UPDATES** . I called support **CUSTOMER SERVICE** multiple times but no one is able to explain why I can't save estimates **ESTIMATES**

3.4 Network Architecture

The sentiment classification process of one or more pre-defined aspects in a review uses two broadly classified learning tasks, that is, sentiment polarity of aspects and document-level sentiment polarity as an auxiliary source. The goal is to narrow down the sentiment polarity that is relevant to the identified aspect, rather than the overall sentiment polarity of the document. Being able to transfer the knowledge from document-level sentiments to aspect level provided better results as it increases generalization during training.

Our architecture consists of three different layers of capsule networks. The first layer extracts features, the second layer learns the semantic construct and the last layer is responsible for sentiment classification. In the next section, we will illustrate the role of every layer in more detail.

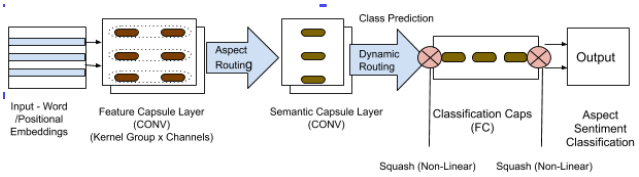

Figure 2

Input Data

There are two types of pre-labeled review data sources that have been used as primary and auxiliary data. The source of aspect-level sentiment data is the QuickBooks mobile reviews and the document-level sentiment data comes from Yelp restaurant reviews.

Input Layer

As part of the data processing for the input layer, we have followed a few typical steps. 1. A word-to-id dictionary was generated from the raw review tokens. 2. Word embeddings were created for all the words in the word-to-id dictionary. The word embeddings are collected from the pre-trained embeddings glove.840B.300d. These embedding vectors are used as the primary input in the network. 3. Positional embeddings of aspects were created and supplied as part of the input. Following [16], relative position values were calculated for all the tokens with respect to the aspect word in a review and additional weights are given to the polarized tokens. Since the

auxiliary data source does not have aspects, there is no need to generate the positional data. So, embedding vectors are filled with zeros. 4. The word and positional embedding vectors are concatenated and put in as feature capsule layer input.

Feature Capsules Layer

In this layer, a set of n-gram feature vectors are extracted from sentence embedding $\boldsymbol{X}$. We then apply multiple convolutions with a fixed number of kernel groups on the input embeddings and it allows us to construct a single capsule of a feature vector $\boldsymbol{r_i}$. For a single sentence, there are multiple capsules that contain a local semantic and contextual meaning of a set of different sliding windows.

$$r_i = X_{i:i+K} * F + b$$

where $F \in R^{dp \times (dh \times K)}$ is the kernel group, $(d_h \times K)$ is the size of one convolutional kernel, K is the n-gram size and d_p is the dimension of one feature capsule.

A capsule network of feature capsules are created after multiple iterations of convolutions with different types of kernel groups. Each of these sets of feature capsules constructs a network of capsule layers. The output vector is constructed from the output of all the channels

$$R = [r_1, r_2, ..., r_C]$$

Aspect Routing

The primary objective of aspect routing is to compute aspect weights in a fixed contextual window. Aspect weight vectors were added with the feature weights from the feature capsule layer and the combined weight vectors become input for the semantic capsule layer.

The aspect routing uses both sentence embedding and the corresponding aspect embedding vectors to generate aspect weights with a sigmoid activation function.

$$a_i = sigmoid(X_{i:i+K} * Fa + T_a e_a + b_a)$$

where X is the sentence embedding, F is the kernel for the convolution, e_a convolution is the aspect embedding and T_a is a transfer matrix. Depending on the output values of the activation function, every aspect receives various weights depending on the context and if an aspect probability score is low, it will not be routed to the next layer in the capsule network. This process is repeated for all the channels in the network.

$$G = [a_1, a_2, ..., a_C],$$

Semantic Capsules Layer

The semantic capsule uses feature capsules with aspect weight vectors as input, then selects max feature probability from each channel to identify the most relevant context to the aspects [18]. Since the max probabilities are chosen from all possible context windows or feature capsules of a sentence, this selection takes into account the entire sentence, rather than a portion (n-gram) of the sentence.

$$P = feature\ capsules\ (R) * aspect\ weights\ (G)$$

After accumulating semantically most relevant features, according to [17], the length of semantic capsules is calculated to represent the probability. A squash function (non-linear activation) is applied.

$$u_i \leftarrow ||u_i||^2\, u_i / (1 + ||u_i||^2\, ||u_i||)$$

From the squash function above, it is clear that vectors with bigger lengths shrink to 1 or close to 1, while vectors with smaller lengths (less than 3-4) get close to 0 probabilities.

Classification Capsules Network

Once the semantic capsules are generated in the previous layer, those go through multiple transformation processes before reaching the fully connected classification network. First, a semantic capsule with a weight matrix is converted to a prediction vector targeting a class capsule.

$$u_{i|j} = W_{ij}\, u_i$$

Here u_i is the vector representation of the semantic capsule and the W_{ij} is a weight matrix. Once all the semantic capsules become prediction vectors, dynamic routing helps to aggregate them and create class capsules. Then the class capsules go through a squash function [17] to get a probability score bounded in [1,0].

4. Experiments and Results

In these experiments, we used Yelp reviews [19] to transfer document-level knowledge. We ran all experiments with a learning rate of 0.001 and batch size 128, and the maximum iterations 66.

We classified our data set by size, that is, reviews with 100 or less words, 200 words or less, and 400 words or less. Running the network with these datasets revealed that we score the best F1 scores with reviews of 200 words or less. The average F1 score for a sentence with 100 words or less was 0.735, whereas it was 0.7349 for 200 words or less, and 0.6811 for 400 words or less. Given that the average review size is less than 200 words, this made us decide to focus our optimizations and experimentations on the reviews with 200 words or less. See figure 3 for the various runs that we captured.

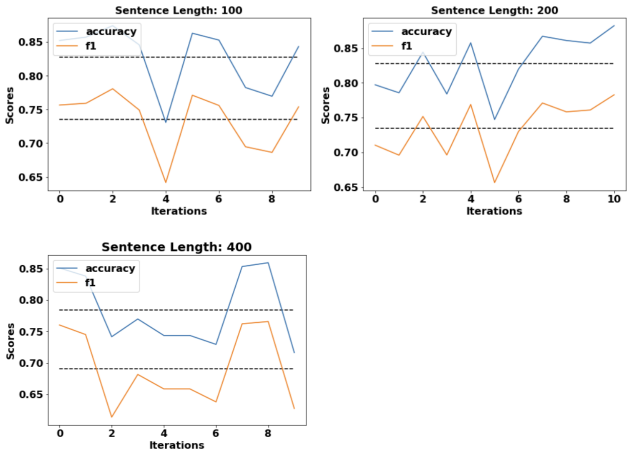

Figure 3

Next, we looked into ways to improve the accuracy of the network. So we explored a couple of approaches to initializing layer weights in the neural network. We compared between Random Uniform Initializer and Xavier (Glorot) Uniform Initializer. Using Random Uniform Initializer, weights are generated with a uniform distribution, whereas in Xavier, the initializer is designed to keep the scale of the gradients roughly the same in all layers. We received better results with Xavier Uniform Initializer. The average accuracy and FI scores were 0.8545 and 0.7604 respectively, compared to 0.8276 and 0.7349 with Random Uniform Initializer. Figure 4 captures scores over 10 runs.

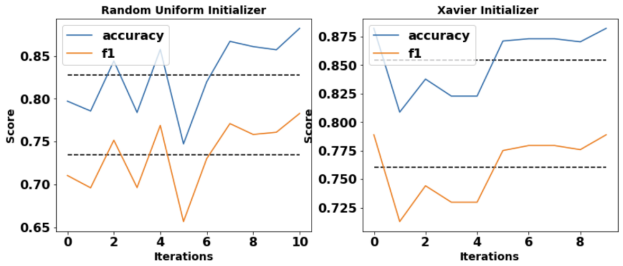

Figure 4

Finally, we wanted to test the effect of positional embedding of tokens and whether we could achieve better scores if we could apply weights that vary by 1. The relative positional index of the token from the aspect, 2. How polarized the word is. In our initial algorithm, the process of creating vectors with relative positional indexes does not distinguish highly polarized words from any less significant words, that is, words with sentiment score of 0. Therefore, we introduced a modification that applies more weights to highly polarized words which are close to the aspect word. For example, a review, such as 'I **love** the **app**, but frequent updates are **annoying**' produces a vector, [4, 3, 2, 1, 2, 3, 4, 5, 6] without considering the weights for the polarized words. However, with added weights for the polarized words '**love**' and '**annoying**' the vector becomes [4, 0.005, 2, 1, 2, 3, 4, 5, 0.01]. To calculate the weights, we are using a formula $w_i = (d_i / s) / c$, where d_i is the distance from the aspect, s is the sentiment score and c is a tunable constant. With these modified embedding vectors, we received an average F1 score of 0.84425 compared to 0.834.

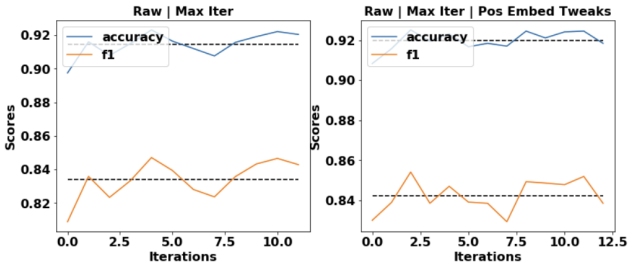

Figure 5

Figure 6 shows the best performing model while training. After the initial choppy start, the model begins to learn and smoothly gains accuracy and f1 scores while reducing loss.

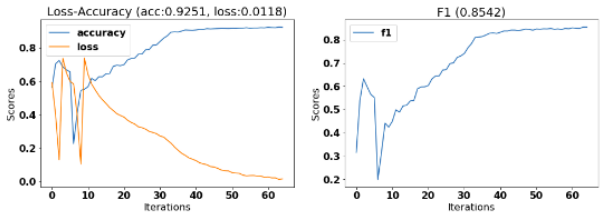

Figure 6

5. Conclusion

Initially, we had the challenge of identifying relevant aspects and labeling of existing review data accordingly. After labeling a good number of reviews, iterative aspect training and modeling helped to overcome the data availability issue. With these labeled records, training the aspect-based sentiment model became a sustainable process.

Aspect based sentiment analysis with Capsule Network goes far beyond the typical document-level sentiment scoring. Neurons in each capsule pick up subtle features or patterns from sentence and position embeddings. Consequently, the layer of networks can make much more informed decisions while classifying.

Since the sentiment scores are based on carefully curated aspects, the model output will profoundly change the way product teams analyze incoming reviews. Further, this aspect-level sentiment analysis will create high-value business action plans for product teams and will influence and guide product teams and leaders of features that require additional focus.

6. References

[1] Mori Rimon (2004), “Sentiment Classification: Linguistic and Non-linguistic Issues”, pp 444-446.

[2] Alec Go (2005), “Twitter Sentiment Classification using Distant Supervision”, Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), Vancouver, CA.

[3] P.Kalaivani, Dr.K.L.Shunmuganathan, “Sentiment classification of movie review by supervised machine learning approach”, Indian Journal of Computer Science and Engineering (IJCSE) ,Vol. 4 No.4, Aug-Sep 2013.

[4] Bing Liu, "Exploring User Opinions in Recommender Systems", Proceeding of the second KDD workshop on Large Scale Recommender Systems and the Netflix Prize Competition", Aug 24, 2008, Las Vegas, Nevada, USA.

[5] Toh Z. & Wang W. (2014) DLIREC: Aspect Term Extraction and Term Polarity Classification System, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) pp. 235 - 240, Dublin, Ireland, August 23-24, 2014.

[6] Liu, B. (2015). Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press.

[7] Nasukawa, Tetsuya and Jeonghee Yi, 2003, Sentiment analysis: capturing favourability using natural language processing, Proceedings of the K-CAP03, 2nd International Conference on knowledge capture.

[8] Dave et al, 2003, Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews, In Proceedings of the 12th International Conference on World Wide Web, WWW 2003, 519-528.

[9] Minqing Hu and Bing Liu (2004). Mining and summarizing customer reviews. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. Seattle, WA, USA, ACM: 168-177.

[10] Alqaryouti et al, Aspect-based sentiment analysis using smart government review data, Applied Computing and Informatics, 2019.

[11] Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. 2009. “Multi-Facet Rating of Product Reviews.” In Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 485–96.

[12] Schouten, Kim & Weijde, Onne & Frasincar, Flavius & Dekker, Rommert. (2017). Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis With Co-Occurrence Data. IEEE Transactions on Cybernetics. PP. 1-13. 10.1109/TCYB.2017.2688801.

[13] Schouten, Kim & Frasincar, Flavius. (2015). Survey on Aspect-Level Sentiment Analysis. IEEE Transactions on Knowledge and Data Engineering. 28. 1-1. 10.1109/TKDE.2015.2485209.

[14] https://quickbooks.intuit.com

[15] Zhuang Chen, Tieyun Qian (2019) Transfer capsule network for aspect level sentiment classification. In: Proceedings of the 57th annual meeting of the Association for Computational Linguistics, pp 547–556

[16] Shuqin Gu, Lipeng Zhang, Yuexian Hou, and Yin Song. 2018. A position-aware bidirectional attention network for aspect-level sentiment analysis. In Conference on Computational Linguistics (COLING 2018).

[17] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In Con- ference on Neural Information Processing Systems (NIPS 2017).

[18] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In AAAI Conference on Artificial Intelligence (AAAI 2015).

[19] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In Conference on Neural Information Processing Systems (NIPS 2015).