Natalie's Little Helper
Automated Twitter Responses

Megan Bull mkbull@usc.edu
Jonathan Lal jlal@usc.edu
Josh Mazen mazen@usc.edu
Carlos Osorio osoriova@usc.edu
Bozhena Pokorny pokorny@usc.edu

1 Introduction

Social media has marked an interesting transition for business-customer relationships. By mentioning a business, platforms like Twitter have become customer forums to share experiences, reviews, complaints, and requests. By engaging and generating content for consumers, brands can generate web traffic, ideally ending with economic success. Generating appropriate responses to a massive influx of comments can be tedious. Thus, we propose an automated comment analysis and response tool for professional social media accounts.

Analyzing Twitter interactions between airlines and their customers will serve as our case study. These interactions range from casual and engagement-based to complaints and customer dissatisfaction. As such, a response is dependent on the emotion (sentiment) and intent expressed by the customer. For example, an airline must identify dissatisfaction and the main complaint within the tweet to respond to a ticketing issue.

During peak travel times and severe travel disruptions, airline customer service lines can be overloaded, making it difficult for customers to receive assistance. People have begun utilizing platforms like Twitter to request service or express disappointment. By creating a tweetbot for customer satisfaction, we can mitigate customers needing support through regular service channels.

2 Related Work

Customer service and satisfaction are critical to the success of a business. Many companies have begun investing in Natural Language Processing (NLP) solutions to drive customer service processes, particularly sentiment analysis, automatic response generators, and conversational agents. For example, IBM developed code patterns to automate customer support emails[3]. Their code patterns utilize NLP models to analyze customer intent via customer service emails and send back generated intent-specific responses.

IBM Watson, a question-answering system, utilizes Natural Language Classifier (NLC) and Natural Language Understanding (NLU) models to generate automated email responses to customer service inquiries. NLC and NLU models classify consumer sentiment and detect context-specific entities via email messages. These entities assist in automated and intent-specific responses. The response prompts further input from the customer when more information is required to complete the service task.

Automated response generators historically employ a range of models, from Support Vector Machines (SVM) to Long Short-Term Memory (LSTM), with varied results. In particular, the Universal Language Model Fine-tuning (ULMFiT) exhibits high performance in text classification tasks[4]. Built from an Averaged Stochastic Gradient Descent Weight-Dropped (AWD) LSTM architecture, ULMFiT models use transfer-learning methods to carry out several NLP tasks, such as text classification. The ULMFiT model requires up to 100 times less training data than other text classification models[4]. The model also can capture properties, such as long-term dependencies, for use in downstream processes. The ULMFiT utilizes language and text classification sub-models to complete a given task. The language sub-model is trained on a large general corpus and fine-tuned on a task-specific specialized dataset. This dataset, and the output of the language sub-model, contribute to the development of the classifying sub-model.

3 Method

The corpus for this experiment was the Twitter Airline data set from Kaggle[8], containing the following features and labels of interest, including tweets from users directed at an airline, the ternary

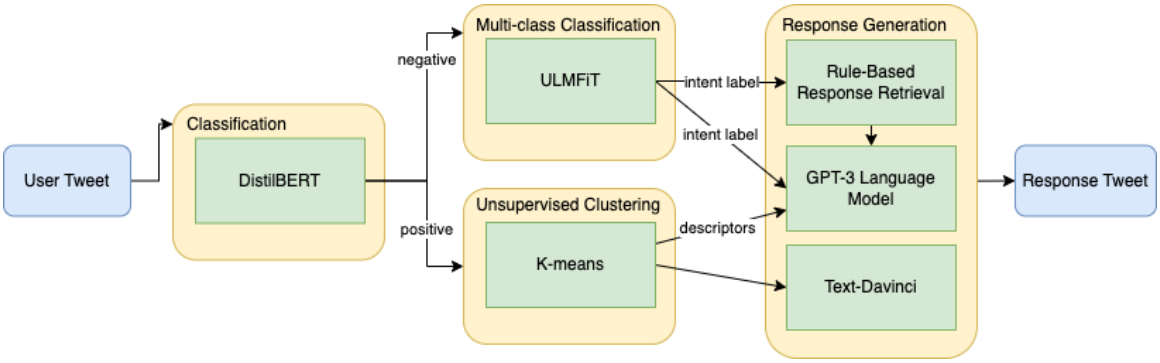

Figure 1: Model Pipeline

sentiment of the tweet, negative reason for a negative sentiment tweet, and confidence scores for sentiment and negative reason labels. We applied contraction resolution, whitespace normalization, lowercasing, and removal of URLs, HTML tags, and non-alphabetical characters for data cleaning. We implemented stop-word removal and lemmatization via the NLTK library for pre-processing. TF-IDF vectorization, average word embedding, and attention mask embedding assisted in feature extraction.

We grouped positive/neutral and negative sentiments into binary labels to fine-tune the DistilBERT[10] model for our final sentiment classification. DistilBERT is a smaller and faster version of the BERT model with 95% of the performance of a regular BERT model. The model was fine-tuned on optimal hyperparameters, found by the Hugging Face Trainer class, after which we handled positive/neutral and negative sentiments separately.

We trained an AWD-LSTM classifier using the ULMFiT approach for negative intent classification. Using the Twitter corpora, we fine-tuned a Fast.AI Language Model (LM). We then built a negative sentiment classifier for negative sentiment tweets and annotated intent from our corpora, utilizing the Fast.AI LM. The ULMFiT LM and AWD-LSTM classifier trained cyclically with an annealing learning rate. All layers, except the embedding layer, were frozen to start. Between consectutive cycles, layers were unfrozen and Fast.AI's valley algorithm was performed to find the optimal learning parameter.

Supervised K-means clustering was performed on negative sentiment tweets with labeled intent to prototype the clustering pipeline and evaluate ground-truth labels. Tweet feature vectors were generated through TF-IDF. Dimensionality-reduction techniques, such as Uniform Manifold Approximation and Projection (UMAP) and truncated Singular Value Decomposition (SVD), were applied to feature vectors to overcome dimensionality challenges in BoW representations. UMAP and truncated-SVD served as input, and performance metrics were collected across K-means model initializations provided by the sklearn package (kmeans++, random k-means, minibatch, SVD-initialization).

Unsupervised clustering was performed on positive and neutral sentiment samples to extract meaningful intent and document topics. Because the value of k is unknown in this unsupervised task, the elbow method was applied to find optimal K-values. That is, a balance between achieving a low loss value and a small number of clusters, is identified.

We applied the basic k-means algorithm to feature embedding representations, such as TF-IDF, TF-IDF with SVD, Word2Vec, and Sense2Vec. TF-IDF vectorization served as our baseline. We then performed reduction through SVD with a various number of components. Vectors were generated with both basic and strict pre-processing techniques. Basic pre-processing included the normalization of whitespace, contraction resolution, and lowercase text. Strict pre-processing, employed basic techniques in addition to stop-word removal, URL/HTML removal, and named airline removal.

We utilized Gensim's pre-trained Google News Word2Vec model to generate tweet embeddings, averaging over the component features. Finally, we applied Sense2Vec, a pre-trained model from spaCy that uses Reddit data and aims to improve the embedding of context-specific words by including syntactic dependencies.

The rule-based response generation primarily focused on responses to labeled negative sentiment tweets. For example, the rule-based generator would retrieve a response containing relevant information for finding lost luggage, submitting a customer service complaint, or helpful tips for booking a flight; based on the labeled sentiment. Since our dataset encompassed multiple airline companies, it is important to note that our responses are not airline-specific. Since, in practice, this generator would be used by a specific airline company, ideal responses would only include information relevant to that company. Adding a rule to formulate responses based on the mentioned airline would ultimately be inefficient since the only use case would be for our project. Therefore, we opted to include responses without specific information for any company, and instead, leave space for relevant information to be displayed. We use this response generation as a fallback method to assist customers with general queries when further context cannot be identified.

We also used a language model to analyze sentiment and intent and use such information to create appropriate and unique responses. Response generation for negative tweets utilized the Grounded Open Dialogue Language (GODEL) model. GODEL conversations are pre-tuned from a massive set of Reddit user data and accept instruction, context, and knowledge as input[9].

With the intent, we can filter an appropriate instruction to give the model for its response and provide the cleaned tweet text for context. This information enables appropriate response generation while also providing the model with sensitivity context. For example, the model formulates sympathetic responses to customers with lost luggage complaints.

The Text-Davinci model from OpenAI[11], based on the Generative Pre-trained Transformer 3 (GPT-3), is used for positive-sentiment response. OpenAI employs a sarcastic and conversational chatbot API, Marv, for model usage. Marv constructs a casual and playful tone for positive tweets to further engage with the customers. To utilize Marv, we prefixed and suffixed tweets with a customer token to specify where the chatbot response would begin.

4 Experimental Setup

For the sentiment classification task, we created a Perceptron and Support Vector Classifier (SVC) to construct a baseline for later comparison with more complex models. We then implemented a Feed Forward Neural Network using dropout between fully connected layers and ReLU activation. The use of pre-trained transformer models, such as RoBERTa and DistilBERT from Hugging Face followed. DistilBERT is a pre-trained model for the focus of our research, due to the poor performance of other transformer models, and fine-tuned for sentiment analysis. Similarly, a Perceptron and a Linear SVC served as the baseline for the intent classifier, with the addition of a logistic regression model.

The corpus used was found to have highly unbalanced categories for both annotated sentiment and intent categories. The least populated intent represented 0.8% of the data, while the most populated represented 32%. To contrast this, multiple up-sampling methods, like Synthetic Minority Oversampling Technique (SMOTE), were used on training data. Implementing a binary sentiment classifier separating negative and positive/neutral tweets accommodated the proposed final pipeline. Similarly, intent classifiers were also trained to group intent categories into four broader classes. For the sentiment and intent classification models, both accuracy and F1 scores served as evaluation metrics for model performance. Additionally, perplexity was used to train the ULMFiT-LM.

We utilized the following pipeline to perform unsupervised clustering and evaluation: feature vector embedding generation, perform elbow method to select the optimal k-value, perform k-means clustering, and find the average silhouette score. Then, for each cluster product, the cluster's silhouette scores the top ten words by feature importance, and yields a word cloud of the 100 highest-scoring words. The success of clustering, as measured by silhouette scores, and manual interpretation of the highest-scoring words in the clusters assisted in human analysis of meaningful groups of tweets in the neutral/positive category.

Due to the lack of a reference response dataset, manual inspection of generated tweets served as our evaluation metric.

Model	Accuracy	F1-Score
Perceptron	0.79	0.78
SVC	0.81	0.80
FNN	0.79	0.77
DistilBERT	0.85	0.85
Binary DistilBERT	**0.87**	**0.87**

Table 1: Sentiment Scores

Model	Accuracy	F1-Score
Perceptron	0.64	0.64
Logistic Regression	0.68	0.68
SVC	0.67	0.67
ULMFiT	**0.73**	**0.73**

Table 2: Grouped Intent Scores

5 Results and Discussion

The best-performing simple model for sentiment used a Linear SVC as shown in Table 1. For complex models, the DistilBERT model exhibited the highest performance with an improvement of 5% and an increase of 7% for binary classification as compared to the baseline models.

The ULMFiT model showed the best performance for intent classification, with a 5% improvement over baseline models, as seen in Table 2. Moreover, all classifiers demonstrated around a 10% improvement in their scores when grouping the intents into the four broader classes. However, the utilization of SMOTE for oversampling and balancing categories decreased performance by about 1% on all sentiment and intent classifiers.

While developing other advanced methods to train the classifiers may slightly improve results, we theorize that substantial improvement will come from improving the quality of the training dataset. That is, ensuring balanced categories for sentiment and intent, as well as, increasing the number of accurate annotations for these fields.

Using the labeled portion of the dataset, we performed K-means clustering with the number of labeled groups, K=10, to evaluate the clustering accuracy with ground-truth labels. All attempts produced low scores using evaluation metrics that used the labels as ground-truth to assess quality. However, the silhouette score greatly improved upon applying dimensionality-reduction techniques. We concluded that the clustering algorithm appeared to be clustering on groups not well-described by the annotated labels, focusing instead on the unsupervised task.

Evaluation of clustering results required human assessment to determine meaningfulness, where interpretation of word clouds proved to be more informative than using raw scores. Iterating through various text representations led to the most meaningful improvements in results. During the first iterations, we found removal of airline names disabled cluster formation around these words as distinct feature groups.

TF-IDF with strict pre-processing and SVD-reduction to 30 components led to the best clustering; in terms of both average silhouette value and top-word interpretation, with an average silhouette of 0.414. However, in every clustering approach, the clusters were uneven in size and contained a large cluster of words related to 'thank'. We attribute this to the make-up of sentiment representation in our dataset. That is, many tweets have positive or neutral-sentiment annotations. Intent, however, was not fully captured by the clusters. Although, in some cases clusters provided other information, such as topic or level of causalness, that could inform model selection in the response-generation task. For this reason, we saw the clustering method as having a purpose in our overall pipeline that differed from the original idea of intent-classification.

For example, an intent cluster containing the words: [dm, sent, follow, email, send, help, confirmation] could be described as 'request correspondence' to generate a rule-based response. For the Davinci-Text LM, a cluster containing slang such as: [fleek, fleet, rt, lol, wow, stop, happened, real] could be described as a casual interaction category to generate an appropriate response.

The GODEL model produced somewhat relevant sentences but could not clearly capture the perspective of a brand representative. Given an instruction to reschedule a flight, it would ask questions related to the airline. We conclude this is likely due to the model not being fine-tuned with customer service data. Further work would aim to create such a dataset and populate this model to create a more natural dialog flow. The Text-Davinci model produced natural, relevant responses for positive sentiment tweets but for neutral tweets, which were primarily queries, the response was not grounded in truth. This is left open for future work to incorporate a knowledge base to reflect company policy.

References

[1] Sridhar Ramaswamy and Natalie DeClerck. "Customer Perception Analysis Using Deep Learning and NLP" *Procedia Computer Science*, 140(18):170–178, 2018.

[2] Kumar, S., Zymbler, M. "A machine learning approach to analyze customer satisfaction from airline tweets." *J Big Data*. 6(19):62, 2019.

[3] IBM. "IBM/Smart-Email-Support: SMART Email Support for Telecom Organisations - Provide Automated Customer Support for Emails." *GitHub, IBM*. https://github.com/IBM/smart-email-support.

[4] Howard, Jeremy, and Sebastian Ruder. "Universal Language Model Fine-Tuning for Text Classification." *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 23 May 2018, https://doi.org/10.18653/v1/p18-1031.

[5] Ankit, and Nabizath Saleena. "An Ensemble Classification System for Twitter Sentiment Analysis." *Procedia Computer Science, vol. 132, 2018, pp. 937–946.*. https://doi.org/10.1016/j.procs.2018.05.109.

[6] Shu T, Wang Z, Lin L, Jia H, Zhou J.. "Customer Perceived Risk Measurement with NLP Method in Electric Vehicles Consumption Market: Empirical Study from China." *Energies (Basel)*, 15(5):1637, 2022.

[7] Jurofsky D, Martin, J.. "Chapter 24: Chatbots Dialogue Systems." *Speech and Language Processing*, 2021.

[8] Data for Everyone Library, CrowdFlower. "Twitter US Airline Sentiment." https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment.

[9] Peng, B., Galley, M., He, P., Brockett, C., Liden, L., Nouri, E., Gao, J. "GODEL: Large-Scale Pre-Training for Goal-Directed Dialog." arXiv preprint arXiv:2206.11309., 2022

[10] Sanh, V., Debut, L., Chaumond, J., Wolf, T. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv preprint arXiv:1910.01108., 2019

[11] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... Amodei, D. "Language models are few-shot learners." arXiv preprint arXiv:1910.01108., 2019 Advances in neural information processing systems, 33, 1877-1901.

6 Division of Labor

- Building, cleaning initial training dataset for sentiment analysis: Jonny/Bozhie/Carlos
- Train classification model for sentiment analysis: Jonny
- Validation and benchmarking of sentiment analysis model: Jonny
- Train classification model for intent analysis of negative tweets: Carlos/Jonny
- Validation and benchmarking of intent analysis model: Carlos
- Exploring options for positive sentiment: Bozhie
- Unsupervised clustering methods: Bozhie
- Rule-based Response-retrieval: Megan
- Language model negative sentiment response generation: Josh
- Language model positive/neutral sentiment response generation: Jonny

Link to our GitHub:
https://github.com/jonnylal/natalies-little-helper