EasyChair Preprint

№ 8562

Smart Customer Care: Scraping Social Media to Predict Customer Satisfaction in Egypt Using Machine Learning Models

Mohamed Anwar, Karim Omar, Ahmed Abbas,
Fakhreldin Abdelmonim, Mohamed Refaie, Walaa Medhat,
Aly Abdelrazek, Yomna Eid and Eman Gawish

August 3, 2022

Smart Customer Care: Scraping Social Media to Predict Customer Satisfaction in Egypt Using Machine Learning Models

Mohamed Anwar [*M.Hosam@nu.edu.eg*], Karim Omar [*K.Omar@nu.edu.eg*], Ahmed Abbas [*Ahm.Abbas@nu.edu.eg*], Fakhreldin Abdelmonim [*F.Abdelmonim@nu.edu.eg*], Mohamed Refaie [*M.Refaie@nu.edu.eg*], Walaa Medhat [*WMedhat@nu.edu.eg*], Aly Abdelrazek [*Al.Razek@nu.edu.eg*], Yomna Eid [*Y Eid@nu.edu.eg*], and Eman Gawish [*EGawish@nu.edu.eg*]

Nile University, Sheikh Zayed City, Giza 3247010, Egypt

Abstract - This paper proposes the utilization of posts from social media to extract and analyze customer opinions and sentiments towards any specified topic in Egypt. Summarized statistics and sentiment values are then displayed to the consumer (companies such as Vodafone, WE etc..) through both an attractive and functional user interface. Text, location, and time of thousands of posts are scrapped, stored, preprocessed, then managed through topic modelling to infer all the hidden themes and delivered to a Recurrent Neural Network (RNN) to output whether the topic was positive or negative. Topic modelling was implemented using the BERT architecture and AraBert word embedding. Sentiment analysis model training was conducted on approximately 4000 rows of processed data and made use of Arabic glove embedding to speed up sentiment and word pattern recognition. Five models were experimented on: LSTM, GRU, CNN, LSTM + CNN and GRU + CNN. Overall, the GRU was the model with the best results, concluding with an accuracy of (86.19%), loss of (0.3349) and an F1-score of (0.858) when validating through the test data.

Keywords—Customer satisfaction, RNN, CNN, LSTM, GRU, Arabic language, Egypt, Social Media, Sentiment analysis, Topic Modelling

1 INTRODUCTION

Customer loyalty is one of the most essential factors in the success of all services and products. Traditional methods of collection such as, surveys, text messages and phone calls have mostly proven ineffective and inconvenient to implement and interact with. This is especially true when considering the lack of tools available to make use of the Arabic language.

Most of the work currently available in the Egyptian customer care space is either limited, outdated, or privately used by giant corporations. (Smart Customer Care) – our project aims to fill that gap by obtaining real opinions and sentiments of products in Egypt while simultaneously providing it in a way that is user friendly, flexible, and adaptive to the Arabic native tongue.

This will allow business owners, to easily obtain relevant and up to date feedback on their products and discover trends in the market, quickly and efficiently.

Once a criterion is described through our web-application, posts related to the selected topic are scrapped from social media. The reason social media was selected as the main source of data collection is due to it being a medium that people worldwide use to express themselves. People also tend to be more honest when not under the stress of face-to-face confrontations. However, the most significant reason is the provided accessibility. Almost everyone uses social media to interact with world for very little to no costs at all. This can ensure a vast amount of data that contains opinions from different communities, cultures, and class levels which overall, reduces any bias that could present itself.

The scrapped data then goes through multiple preprocessing cycles to remove any unnecessary variables. This clean data is then handled by our sentiment analysis machine learning model. It predicts whether the selected topic is positive or negative and stores the results in a cloud database to be accessed from anywhere. After the full analysis, sentiment recognition and compilation of our acquired data, everything is displayed to the user through a responsive and interactive web - user interface. An account authentication system is utilized alongside the interface to save any custom settings and keywords. Users will also have access to a dashboard that will act as a central hub from which they can utilize all the features present in the application.

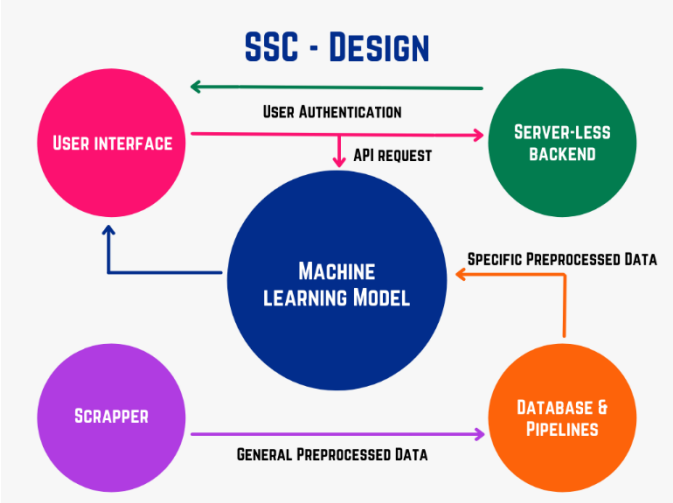

Figure 1. Project Modules

The main contributions present in this project include but are not limited to focus on creating tools that targets the Arabic market and language. As there is a relevant lack of applications and technology in this space. In addition to that, data collection methods (will be illustrated in later sections) are achieved through custom made web scrappers and unrestricted tools. This allows for genuine data collection and reduced bias as everything is mostly under our control. This application overall provides companies feedback, sentiment values and the topics being discussed by the people using their products, all through a responsive and interactive user interface.

Regarding the upcoming sections, many of the ideas and relevant discoveries will be discussed in detail.

Section 2: This section's purpose aims to explain the different neural network implementations and model training techniques that are crucial to understanding how our implemented models work and function. The literature review is the subsequent segment, where we discuss projects and research related to our work, that was previously conducted in the past to provide insights on how to move forward with this project. **Section 3:** Our implementation is then explained by how each model was trained and tested, data collection methods, where and how data is stored and preprocessed into the format needed by the models. Followed

by a dedicated section describing how the project's modules come together to form a fully functional web application. **Section 4:** This section focuses on the results obtained from training the sentiment analysis models and the different scraping techniques implemented. Which directly influence the decisions to be made moving forward with the project. **Section 5 & 6:** The final two section concentrate on the conclusion and future work which relays, summarizing the project stages, methods used, and corresponding results, in addition to useful implementations not utilized in this project but can be taken advantage of in the future.

2 LITERATURE REVIEW

Work done that relates to smart customer care will be discussed in this section. Marketers use emotion mining in customer care services to learn about client happiness and how to improve their service or product. Additionally, user emotions can be used to forecast sales of a certain product. Recently, the benefits and uses of analyzing emotions in Twitter posts are numerous. These include e-commerce, e-marketing, and other uses. Some of the short-term goals that businesses have set for themselves include tracking their internet reputation as well as learning how people feel about pertinent goods, services, events, or people. Few researchers dive deep to evaluate and classify the emotions behind tweets, particularly in Arabic tweets, while most studies concentrate on sentiment analysis as positive and negative. When it comes to the work done in this field that handles Arabic data, there is a variation in the approaches to smart customer care.

2.1 Sentiment Analysis Models

In Emotions extraction from Arabic tweets, the authors' way to conduct sentiment analysis was not limited to positive or negative classification. Instead, the authors of the work presented in the paper Emotions extraction from Arabic tweets "constructed a model to extract and classify customer emotions from Arabic tweets based on four feelings: sadness, joy disgust and anger" [24]. They conclude that the experimental outcomes show the viability of the suggested model, which advances the state of the art in the categorization of Arabic tweets using support vector machines (SVM) and naive bayes (NB), which produce the best outcomes. SVM outperforms the other employed classifiers with an accuracy of 80.6 percent, while the NB outperforms the others with an ROC area of
0.95.

On their paper "CNN for situations understanding based on sentiment analysis of twitter data", the authors proposed a different approach to apply sentiment analysis. Their approach is to understand situations in the real world through

sentiment analysis of a dataset of twitter tweets. They explained their approach by saying that "the biggest reason to adopt CNN in image analysis and classification is because CNN can extract an area of features from global information, and it is able to consider the relationship among these features. The above solution can achieve a higher accuracy in analysis and classification" [25]. The logic behind their approach lies in the fact that a CNN's convolutional layer is capable of extracting information from a large piece of text, so they designed a CNN model and tested it on benchmark values (standard values) to compare its accuracy to other traditional methods. The results show that their proposed model gave better accuracy than some of the traditional sentiment analysis methods like the SVM and Na¨ıve Bayes methods.

To achieve multi-classification for text sentiment, Li and Qian proposed a RNN language model, specifically a Long-Short-Term-Memory (LSTM) model, which they claimed can get complete sequence information efficiently, since compared to a traditional RNN model LSTM is better at analyzing emotions of long sentences [26]. They trained different emotion models using the same dataset to conclude which emotion does each sentence belong to. Their experiments resulted in the conclusion that LSTM can produce higher accuracy and recall rates than conventional RNNs.

A survey on sentiment analysis is provided by Pang and Lee. The authors focus on applications of sentiment analysis that go beyond extracting a sentiment value from a single text. Their applications range from sentiment computation towards identifying topics of a text, the visualization of sentiments as well as automatically defining the usefulness of a customer review. Our focus is on methods that seek to address the new challenges raised by sentiment-aware applications, as compared to those that are already present in more traditional fact-based analysis. [27]

A different method is proposed by Ray and Chakrabarti as their approach is based on using R software which can use twitter API to apply sentiment analysis to the users of twitter. Their approach is mainly based on three steps, which are collection of data, preprocessing of data, and then applying actual sentiment analysis using a lexicon-based approach [28].

2.2 Topic Modeling

In the paper smart literature review, they developed a topic modelling algorithm that automatically builds a literature review by extracting topics from a given paper then collecting other available relevant papers. The papers are first preprocessed and prepared for the model to achieve good analytical results. Once the papers have been cleaned and a decision has been made on the number of topics, an LDA method is run. "The outcome of the model is a list of papers, a list

of probabilities for each paper for each topic, and a list of the most frequent words for each topic." [29]

In Improving Arabic Cognitive Distortion Classification in Twitter using BERTopic, they developed a topic modeling method using BERTopic to improve the detection of biased thinking pattern in text -cognitive distortions- to help users "restructure how to perceive thoughts in a healthier way" [30]. This paper proposed a machine learning-based approach to improve cognitive distortions' classification of the Arabic content over Twitter which had several problems, such as text shortness which makes it very hard to find co-occurred patterns and a lack of context information. The topic modeling method they developed enriches text representation by defining the latent topics within tweets. The results demonstrated that their approach outperformed the baseline models. It proved that using latent topic distribution, obtained from the BERTopic technique, can improve the classifier's ability to distinguish between different cognitive distortions categories.

3 METHODOLOGY

3.1 Dataflow

Figure 3 illustrates the flow of data from when data scrapping starts until it reaches the user through the web user interface.

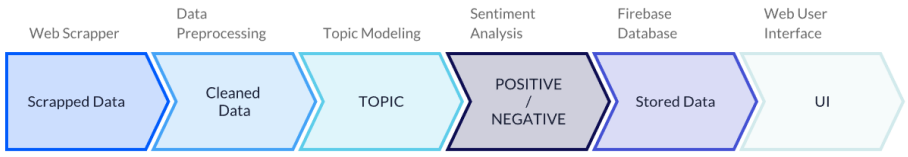

Figure 3. Dataflow Process from Scraping to UI

Data is first scrapped from social media (Facebook and Twitter) followed by cleaning of the collected data from unnecessary features that bias any outputs. Topic modelling and sentiment analysis are individually applied, and their results

are stored in the cloud database. The web application retrieves the data and displays it to the user through the front-end user interface.

3.2 Sentiment Analysis Models

Choosing the most appropriate sentiment analysis model to build was a key factor in the decision-making process of this project. The final judgment was to experiment with varying implementations of Recurrent Neural Networks (RNN), Due to the RNNs ability to remember different relationships between words, they perform exceptionally well when dealing with long sequences of words.

As well as Convolution Neural Networks (CNN), as they optimally perform well when extracting meaning from single words without giving priority to order or sequence.

Combination of Both Sentiment predictions would benefit greatly by combining both the single word and word sequencing abilities both model architectures provide. [31]

Model Libraries: Initially, at the start of development it was decided that cloud-based hardware services would be avoided in favor of local machine resources. This decision held significant priority to allow for offline development and to avoid data loss complications due to internet outages while training. TensorFlow's Keras framework in combination with VS-Code's compilation and editing software was used for the actual implementation of the model. This is due to possessing easy to access tools and functions not readily available through other libraries.

Training Dataset: To create the training data, we utilized Kaggle's vast array of dataset libraries. From which we used a set from 2019 consisting of real Twitter tweets and their corresponding sentiment. The set was split into 2436 positive represented by (1) and 1816 negative tweets represented by (0) which is slightly skewed towards the positive but will be taken into consideration when building the model architecture. After going through the same preprocessing stages mentioned in the previous sections, we overall concluded with approximately 4000 rows of trainable data.

Processing Tokenization: Before using the data for model training some processing and techniques were applied to substantially improve model accuracy, avoid fitting issues while training and to restructure the data into a format comprehensible by the model. The dataset was first randomly shuffled to reduce the risk of the model overfitting and bias towards a specific sentiment. Tokenization of the data was then implemented. This is the process of breaking up

sentences into separate phrases or words each representing some sort of meaning or significance. Encoding of the tokens was later applied to convert them into a unique series of vectorized numbers. The data then went through a padding and truncating process, where any token with a length below or above a specified threshold would be padded with zeros or dropped at the threshold mark respectively. This was all to prep the data for training as the models require encoded data that are all the same length to operate efficiently. In addition, in this case the threshold with the most optimal results was the average word length of all tweets used in training. The data was then split into train and test segments using eighty percent from the data set as training data and twenty percent for testing.

GloVe Embedding: Training and word embedding was applied to the model using the correlations available in a compiled 256-dimension GloVe file of the Arabic language. Parsing of the file was next in this process where the words and its corresponding 256-dimension vectors were extracted and formatted into a python dictionary. Simultaneously while extracting, a condition would check if the obtained Arabic word existed in the original train data. This is to ensure no irrelevant variables will skew the model training while at the same time, wasting utilized processing power and time.

3.3 Model Architecture

Recurrent Neural Network: In the RNN implementation both a GRU and LSTM cell approach was used to create the model. The GRU uses low memory, is fast and well known for excellent convergence with small datasets. Now even though the LSTM is not the best when dealing with low amounts of data, it provides more stable infrastructure when used in training. Therefore, considering the above use cases these types were the most appropriate options to consider.

Convolution Neural Network: The CNN implementation utilized 1D (dimension) convolution layers alongside max pooling, which reduced the size of the input and only sent the important data to the next layer. This helped reduce the number of computations in the network and provided lower probability of overfitting occurrence.

Combinations Between CNN, LSTM and GRU: Two combinations were tested between all 3 model types, CNN + LSTM and CNN + GRU. All the parameters and layers used were identical to the individual model components. This is due to achieving the best results when tested with different architectures and hyper parameters.

To store all the preprocessed data for online usage by our front-end user interface and model training, Firebase-Firestore cloud storage was made use of. [32] Firestore is a NoSQL type of database among Firebase's many serverless services that avoids the traditional tabular structure in favor of more flexible and dynamic interactions. Otherwise, more time and effort were going to be invested into building entire project schemas and relations, while also being forced to abandon the serverless service provided by Firebase for manual database management. This decision overall allowed for quick and easy deployment of the system.

Data Format: NoSQL databases store data using a concept of collections and documents. Where documents are what contain the actual data and a collection is the header encapsulating it. This combo also only comes in even pairs, for every collection there must exist 1 document. Smart Customer Care segments the storing process into a hierarchy of 4 components, the root collection which defines where the data is coming from, in this case Twitter. The second layer is the document representing the inputted keyword / search criteria of the user, for example, Vodafone, WE etc... Third layer contains another nested collection that defines the date of when the data was scrapped. The final layer is a document that is labelled by the time of day that the data was scrapped and contains the preprocessed data.

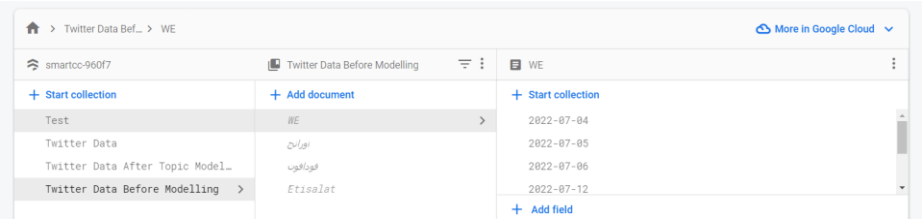

Figure 2. Database Modules

Restrictions: When uploading the data, due to the usage of the provided free plan, there is a 1mb limit per document stored. Scrapped content for any given day is unpredictable and can easily exceed that said limit. To avoid any errors and exceptions, a dynamic implementation in code was used to split the data while uploading into multiple data frames, each with a max size of 2500 rows. Which after testing and comparison was determined to be the optimal number of rows that complied within the size limit while also being in correspondence with the data format.

3.4 Data Collection

Smart Customer Care (SCC) is based on the idea of extracting sentiment out of common people's opinion, and as already established, common people's opinion is mainly voiced through social media, so the first step of this implementation is to collect data from social media. According to the initial plan of SCC, data is to be scrapped off specific social media platforms, these are Facebook, Instagram, and Twitter, starting with which has the greatest userbase, Facebook. A web scraper was developed to scrape Twitter's posts by providing a single keyword, the scraper will then scrape all posts that include this keyword in their text.

Web Scraping: To collect data from websites, a Web Scraper is implemented. According to a report by Oregon State University, "Web scraping is a technique to extract data from the World Wide Web (WWW) and save it to a file system or database for later retrieval or analysis. Commonly, web data is scrapped utilizing Hypertext Transfer Protocol (HTTP) or through a web browser. This is accomplished either manually by a user or automatically by a bot or web crawler". [33]

1) **BeautifulSoup:** A python library called BeautifulSoup is used to create the first web scraper to be utilized into SCC, BeautifulSoup is a Python parsing library that allows you to pull data from HTML or XML pages. But it does not have any crawling capabilities, so a crawler had to be manually implemented in a python script.

2) **Scrapy:** Scrapy is a web crawling and scraping framework, it facilitates crawling various web pages to ben then downloaded, parsed, and stored for later use. The most important feature of scrapy is that it is fast, and can process asynchronous requests, which means it can scrape multiple web pages parallelly.

Facebook Scraper: The developed scraping script crawls through Facebook using its search engine, by passing the desired keyword as an input, the script retrieves all posts which include the keyword, alongside all the data about the post itself (Time of posting, number of likes, number of each reaction, all information about the author, and more) which is later stored as a data frame. However, Facebook proved inefficient for collecting data periodically over extended periods of time, and illegal to use for commercial purposes.

Twitter Scraper: The second social media platform to be scraped is Twitter, this platform offers an array of solutions for scraping data, since it is better built and

more accessible, and its policy of free speech allows for easy and legal ways to collect data. For data to be collected from Twitter, a license was applied for and later acquired, this license allows us to use Twitter's official API (Tweepy) to collect all the data we needed, saving the time needed for developing a new custom scraper. Twitter is the main source of our data since it is consistent and has an abundance of relevant and much needed opinions voiced through it.

Collected Data: The average size of collected data is 200 rows per day, the data set contains 5 columns, "Tweet" the actual text data, "Location" where the tweet is posted from, "year, month, day" the exact date of posting, making the dataset look as follows

Tweet	Location	Year	Month	Day
فودافون رجعت في كلامي انا حره انترنت_غير_محدود_ف_مصر httpstco2Q62hHdkv2	Alexanderia, Egypt	2022	7	8
و الي we يا ريت نلغي الاشتراك من مش هيقدر يقعد من غير انترنت يروح شركة تانية و كمان غير فودافون لازم نأثر فيهم Unlimited_internet_in_egypt	Cairo, Egypt	2022	7	9

Table 1. Collected Data Example

This is an automated process that occurs every day for the convenience of the user.

Data Pre-processing: Scraped data, specially that off social media, often contains numerous unneeded characters such as weird symbols and emojis, all this irrelevant data must be cleaned off before proceeding.

To pass through a sentiment analysis model, language data must be prepared first by removing irrelevant linguistic features that may or not interfere with the result of the model, it differs from one language to another, but some common linguistic features must be removed, like stop words. To remove Arabic stop words, we need a database that contains a classification of all Arabic words and grammar, and for that we used a set of open-source Python tools for Arabic natural language processing.

Before Cleaning	After Cleaning
😣 حرامية!! لسة في انترنت محدود؟ شركةWE #انترنت_غير_محدود_في_مصر	شركةWEحرامية لسة في انترنت محدود انترنت_غير_محدود_في_مصر

Table 2. Data Cleaning Example

Using a method called DefaultTagger, it generates tags for a given feature by first disambiguating a word using a given disambiguator, and then returning the associated value for that feature. It also provides sensible default values when no analyses are generated by the disambiguator or when a feature is not present in the disambiguation. These tags include whether the word is a verb, preposition, conjunction, and more. Using this disambiguator, any word that matches the tags "conj" for conjunction or "prep" for preposition, are removed from the dataset before getting uploaded to the database for later use. [34]

3.5 Topic Modelling

To apply topic modelling, we utilized Google's BERT architecture to identify the topics and aspects of the collected data. The model we specifically used was obtained from the BERTopic python library where we also utilized the library's provided Arabic word embedding functionality called AraBERT, to increase the model's accuracy. The model uses the preprocessed data to discover keywords and trends people are currently interested in.

Company	Topics	Content	Occurrence Frequency
WE	Topic 1	• انترنت_لا_محدود_في_مصر • شركة • الباقة • مقاطعة	69
	Topic 2	• كهربا • الزمالكاويه • التيشرت • بلد • الاهلي	28

Table 3. Topic Modelling Data Example

3.6 Web Application

User Authentication: As previously mentioned, users possess the flexibility to customize their search criteria based on any topic of their choice. To actively save the selected keywords and custom dashboard settings an account registration system was required. To achieve this system an implementation of Firebase Auth was integrated into the application. Using Firebase's API to save new user's credentials into a cloud database and to authenticate any login attempts from the app.

Dashboard: The building of the website structure, code optimization and state management in the user interface was using the ReactJS library. While styling, layout, and support for layout responsiveness for all screen sizes was by utilizing the Bootstrap framework.

Displaying Data: Fast-API was used to communicate between the front-end and back-end modules. A POST http request method was used store the scraping criteria entered by the user which shortly after triggers the scraping process. Same principle applies to retrieving data, a GET http request method is used to read stored preprocessed data which is then compiled into statistical and sentimental records that are displayed on the website.

Data Compilation: The actual stored data in Firebase does not contain any of the aggregation statistics displayed in the application such as, the average and median

word length of the posts. Instead, it is downloaded as a JavaScript object that is parsed to isolate each data column into separate arrays split into sentiment, tweets, and location / time. Using aggregation, reduction, and mapping functions this parsed data is transformed into the final information displayed on the site.

Performance Optimization: To ensure fast loading times and high operation efficiency many optimizations were configured and employed, from which include storing downloaded data from firebase into the browsers local cache to minimize database read operations, which allows for reduced loading times, reduced traffic on the database and lower bandwidth consumption

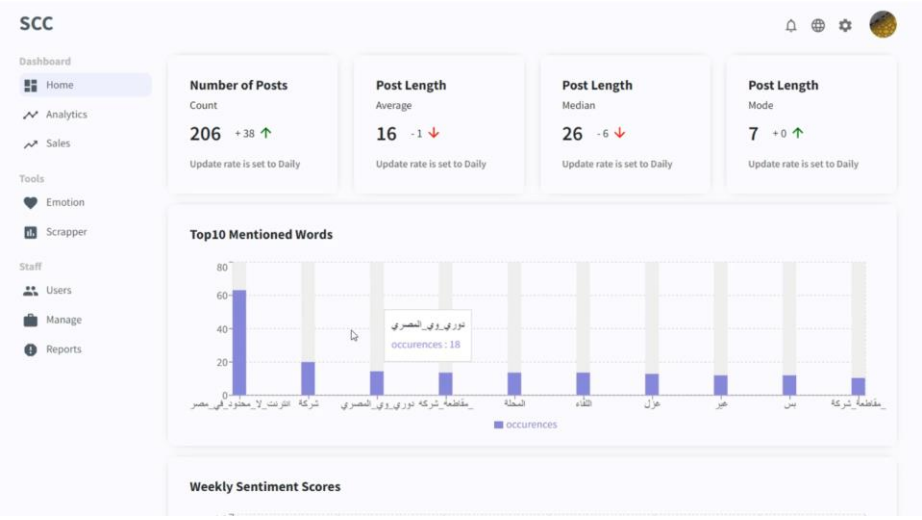

Figure 4. User Dashboard

4 RESULTS

Throughout the training process, many hyper-parameters were tested and experimented with. All the following results in this section utilized the best parameter values that concluded with the highest accuracy outcomes. All models also shared these parameters: binary cross entropy loss function, Adam optimizer,

learning rate of 0.0001, A max sentence length of 20 (the padding and truncating stage), 200 epoch iterations except for the GRU implementation that used 300.

Each result is described through a line graph sketched between accuracy and the iteration count. Followed by a confusion matrix to further elaborate the performance of the classification algorithms used on the test data. As for the numerical scores, the following statistics were obtained: train accuracy, test accuracy, train loss, test Loss, recall, precision and F1 score.

4.1 Architecture Parameters

RNN: We experimented on two RNN models, LSTM and GRU to monitor which one will provide the highest contribution level. Through testing we determined that 20 LSTM cells & 15 GRU cells returned the most optimal results.

CNN: The second type of neural network experimented with was a (CNN) convolutional neural network. In this model we used:

A 1-dimension convolution layer with 16 filters, a stride of 3, “same” padding and a RelU activation function A 1-dimension max pooling layer with a pool size of 4 and stride of 1. Multiple dropout layers with a 20% input and the final layer was flattened and added to a linear dense layer with an output of 1 and a sigmoid activation function.

RNN + CNN: Multiple hyperparameters were adjusted during the testing phase but coincidently the variables with the most optimal results were the same as the standalone models mentioned above.

4.2 Output Results

	Train Accuracy	Test Accuracy	Train Loss	Test Loss	Recall	Precision	F1 Score
LSTM	96.13%	84.40%	0.1092	0.3559	0.837	0.844	0.840
GRU	98.03%	86.19%	0.0610	0.3349	0.856	0.862	0.858
CNN	96.24%	83.93%	0.1103	0.3605	0.832	0.839	0.835
CNN + LSTM	98.66%	80.95%	0.0336	0.4059	0.802	0.807	0.804
CNN + GRU	99.20%	83.21%	0.0236	0.3755	0.828	0.829	0.829

Table 4. Results from Models

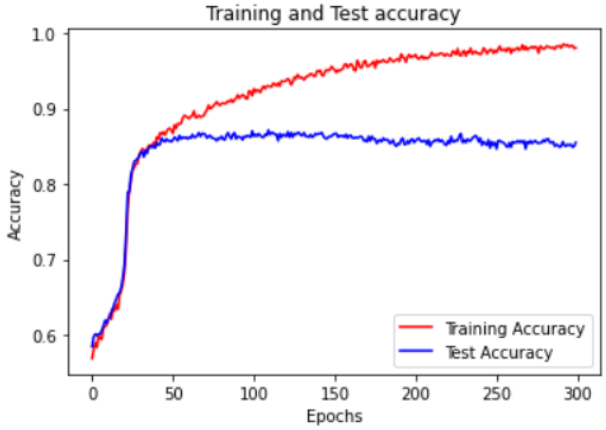

Figure 5. GRU, Accuracy / Epoch Graph

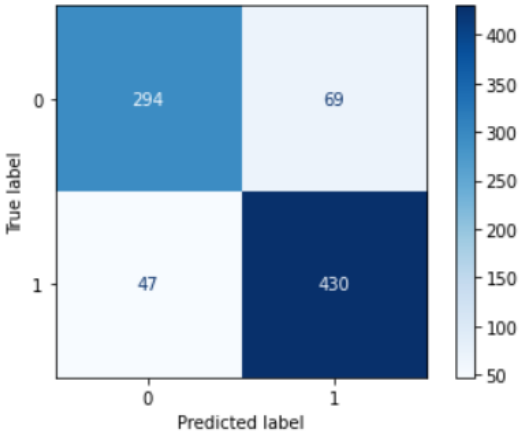

Figure 5.1 GRU, Confusion Matrix Results

Overall, all the models performed similarly when tested but the GRU recurrent architecture under these implementations and parameters scored the highest in all categories, making it our choice of model going forward with this project.

A trend displayed through all the model accuracy graphs was the sudden spike in the beginning of training. The reason behind this was due to the applied GloVe embedding before training. This allowed the training to start from a significantly high accuracy (50%-60%)

Regarding the number of correct and incorrect outputs, the confusion matrix in the GRU model contained slightly more true values than all the other models which is to be expected considering it converged with the highest test accuracy score of 86%

Another revealed trend is regarding bias towards the positive results. The dataset as mentioned in an earlier section contained more positive rows than negative. We were initially afraid of overfitting and bias occurring towards the positive sentiments. However, due to the carful placement of dropout layers, data shuffling and max pooling in the convolution models we have obtained a more stable outcome. This is demonstrated in all the confusion matrices as the number of false results consistently were between 14% – 19%.

4.3 Final Scrapping Methods

The manual crawler for BeautifulSoup worked perfectly but, there was some issues due to it being just an HTML parser, crawling through layers and stacks of nested web pages needed more than average time to complete, taking an average of 5 minutes to scrape 600 Facebook posts (120 posts per minute), therefore this scraping method was deemed inefficient, and a new method had to be used.

When using Scrapy, it scraped an average of 100 posts per second, which is 50 times faster than BeautifulSoup, making it our scrapping library of choice.

5 CONCLUSION

In this project, we built a fully functional and integrated system that scrapes customers' opinions on any topic of choice from social media. Collected data goes through a cleaning process which includes removing stop words, nulls, and other unnecessary features. It is then utilized in topic modelling and sentiment analysis to provide companies with an average of sentiments on their given product, the trends of this sentiment across periods of time, specific statistics on the shape of the collected data and sections each representing a peaking topic or trend in the Egyptian community. All displayed through an interactive user interface. We developed, from scratch, two web scrapers for two social media websites (Facebook & Twitter), multiple data cleaning and preprocessing methods for preparing data. Applied BERT topic modelling to infer the hidden themes in the data. While also testing various sentiment analysis models using different deep learning neural network algorithms, which includes models for LSTM, GRU, CNN, and combinations of the three. The GRU recurrent neural network was selected as SCC's sentiment analysis model as it achieved the highest accuracy score of (86.19%), lowest loss of (0.3349) and F1 score of (0.858).

6 FUTURE WORK

Some work will have to be done for Smart Customer Care to become a commercial product, some of which are already in development. These include but are not limited to Model Improvements: The sentiment analysis model will be rebuilt to achieve higher accuracy and lower loss. Data Preprocessing: Remove more data from text that might interfere with the model accuracy. More Automation: Add an option for the application to run automatically on a preset periodical basis. More

options for the user: A user profile system will be introduced to help users customize scraping periods. Security and Privacy: A better user authentication system will be implemented to ensure the security of user data. Additional details to model outcomes: Implementation of an aspect-based model.

REFERENCES

[1] H. Ana Margarida Gamboa, "Customer loyalty through social networks: Lessons from Zara on Facebook," in Business Horizons, Lisboa, 2014, pp. 709-717.

[2] K. Godson Michael D'silva, "Real World Smart Chatbot for Customer Care using a Software as a Service (SaaS) Architecture," in International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2017.

[3] Z. Lujia Pan, "An Intelligent Customer Care Assistant System for Large-Scale Cellular Network Diagnosis," in International Conference on Knowledge Discovery and Data Mining, 2017.

[4] IBM, "Recurrent Neural Networks," IBM, 14 September 2020. [Online]. Available: https://www.ibm.com/cloud/learn/recurrent-neural-networks. [Accessed June 2022].

[5] I. Jordan, "Machine learning: Trends,," sciencemag, vol. 349, no. 6245, p. 7, 2015.

[6] K. Wakefield, "SAS," [Online]. Available: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html#:~:text=There%20are%20four%20types%20of,%2Dsupervised%2C%20unsupervised%20and%20reinforcement.. [Accessed 6 2022].

[7] P. Baheti, "V7 Labs," 20 June 2022. [Online]. Available: https://www.v7labs.com/blog/supervised-vs-unsupervised-learning. [Accessed July 2022].

[8] "Altexsoft," 18 March 2022. [Online]. Available: https://www.altexsoft.com/blog/semi-supervised-learning/. [Accessed June 2022].

[9] M. Emre Celebi, Unsupervised Learning Algorithms, Shreveport, LA,: Springer International Publishing Switzerland, 2016.

[10] I. C. Education, "IBM," IBM, 21 September 2020. [Online]. Available: https://www.ibm.com/cloud/learn/unsupervised-learning. [Accessed June 2022].

[11] Y. Meng-Hiot Lim, Reinforcement Learning, Singapore: Springer-Verlag Berlin Heidelberg, 2012.

[12] J. Hopfield, "Artificial neural networks," IEEE Circuits and Devices Magazine, vol. 4, no. 5, p. 10, 1988.
[13] M. H. Hassoun, Fundamentals of Artificial Neural Networks, Massachusetts: Massachusetts Institute of Technology, 1995.
[14] B. Y. ARTIFICIAL NEURAL NETWORKS, New Delhi: PHI Learning Pvt. Ltd., 2009.
[15] J. Bouvrie, "Notes on Convolutional Neural Networks," Massachusetts Institute of Technology, Cambridge, 2006.
[16] K. O'Shea, "An Introduction to Convolutional Neural Networks," arXiv, Ceredigion, 2015.
[17] G. C´ıcero Nogueira dos Santos, "Deep Convolutional Neural Networks for," in The 25th International Conference on Computational Linguistics, Dublin, 2014.
[18] Medsker, Recurrent neural networks: design and applications, London: CRC press, 1999.
[19] H. Schmidhuber, "Long Short-Term Memory," MIT Press, 1997.
[20] W. Zaremba, "RECURRENT NEURAL NETWORK," in ICLR, New York, 2015.
[21] K. Chowdhary, "Natural Language Processing," in Fundamentals of Artificial Intelligence, New Delhi, Springer, New Delhi, 2020.
[22] IBM, "Natural Language Processing (NLP)," IBM, 2 July 2020. [Online]. Available: https://www.ibm.com/cloud/learn/natural-language-processing. [Accessed June 2022].
[23] Pelevina, "Making sense of word embeddings," arXiv , Harvard, 2017.
[24] Manal Abdullah, Muna Almasawa, Ibtinhal Makki, Maha Alsolmi, Samar Mahours, "Emotions extraction from Arabic tweets," 2017. [Online].
[25] C. Shiyang Liao, "CNN for situations understanding based on sentiment analysis of twitter data," Procedia Computer Science, vol. 111, pp. 376-381, 2017.
[26] J. Q. Dan Li, "Text sentiment analysis based on long short-term memory," 2016.
[27] L. Bo Pang, "Opinion mining and sentiment analysis," Opinion mining and sentiment analysis, 2008.
[28] P. R. Amlan Chakrabarti, "Twitter sentiment analysis for product review using lexicon method," 2017.
[29] C. Asmussen, "Smart literature review: a practical topic modelling approach to exploratory literature review," Big Data, vol. 6, no. 93, p. 18, 2019.

[30] Alhaj, F., Al-Haj, A., Sharieh, A. and Jabri, R., 2022. Improving Arabic Cognitive Distortion Classification in Twitter using BERTopic. International Journal of Advanced Computer Science and Applications, 13(1).

[31] Alayba, A., Palade, V., England, M. and Iqbal, R., 2018. A Combined CNN and LSTM Model for Arabic Sentiment Analysis. *Lecture Notes in Computer Science*, pp.179-191.

[32] Firebase. n.d. Firestore | Firebase. [online] Available at: <https://firebase.google.com/docs/firestore> [Accessed 31 July 2022].

[33] B. Zhao, "Web Scraping," Encyclopedia of big data, Oregon, 2017..

[34] Tedboy.github.io. n.d. tedboy.github.io. [online] Available at: <https://tedboy.github.io/> [Accessed 31 July 2022]

[35] He Zhao, "Topic Modelling Meets Deep Neural Networks: A Survey," arXiv, 2021.