Using Big Data and Serverless Architecture to Follow the Emotional Response to the COVID-19 Pandemic in Mexico

Edgar León-Sandoval(✉), Mahdi Zareei, Liliana Ibeth Barbosa-Santillán, and Luis Eduardo Falcón Morales

School of Engineering and Sciences, Monterrey Institute of Technology and Higher Education, Monterrey, Mexico
leon.s.edgar@tec.mx

Abstract. The emergence of the COVID-19 pandemic has led to an unprecedented change in the lifestyle routines of millions of people. Beyond the multiple repercussions of the pandemic, we are also facing significant challenges in the population's mental health and health programs. Typical techniques to measure the population's mental health are semiautomatic. Social media allow us to know habits and daily life, making this data a rich silo for understanding emotional and mental well-being. This study aims to build a resilient and flexible system that allows us to track and measure the sentiment changes of a given population, in our case, the Mexican people, in response to the COVID-19 pandemic. We built an extensive data system utilizing modern cloud-based serverless architectures to analyze 760,064,879 public domain tweets collected from a public access repository to examine the collective shifts in the general mood about the pandemic evolution, news cycles, and governmental policies using open sentiment analysis tools. We provide metrics, advantages, and challenges of developing serverless cloud-based architectures for a natural language processing project of a large magnitude.

Keywords: Sentiment analysis · Big data · COVID-19 · Machine learning · Mexico · Twitter

1 Introduction

On February 27, 2020, Hugo López-Gatell Ramírez, the head of the Undersecretaries of Prevention and Health Promotion at the Mexican Secretariat of Health, reported a patient in the INER (National Institute of Respiratory Diseases) as the first official case of COVID-19 reported nationwide [21]. From there, a general lockdown was mandated on April 21, 2020. Since then, the country has followed actions based on non-pharmaceutical interventions (NPI) to mitigate the effects of the pandemic on the general population. However, the COVID-19 pandemic also challenges individuals' emotional and psychological well-being.

P. Navaux et al. (Eds.): CARLA 2022, CCIS 1660, pp. 145–159, 2022.
https://doi.org/10.1007/978-3-031-23821-5_11

This challenge raises the need to incorporate emotional health-related data into organizations' decision-making processes and to build the appropriate dashboards showing critical information. Even though it is clear the importance of tracking the day-to-day data on the pandemic progression regarding ongoing infection rates and fatality, among other statistics, another important dimension, emotional health, needs a proper measuring instrument. These reasons point to the need to build, deploy, and maintain a large-scale, resilient system capable of performing sentiment analysis in large, continuous data sets, such as ongoing Twitter traffic.

Acquiring and processing this amount of information is not an easy task, as this is a task that presents all of the challenges described by the three Vs. of Big-Data: Volume, Variety, and Velocity [15]. For large-scale sentiment analysis systems, these challenges are explored in depth by [5] adopting a broader definition of big data. There are multiple definitions containing different aspects of these architectures, such as analysis, value, computer power, visualization, variability, and integrity, among others. An in-depth description of these definitions is described by [28], concluding that the challenges presented depend on the context of the task. However, as already defended by [15], this problem adheres to the extensive data domain in multiple dimensions beyond the original three dimensions proposed. Several challenges must be addressed as a big data problem, which is listed next.

1. Acquiring the emotional data of a large population of individuals, either in a traditional methodology or through social media, can be costly.
2. Processing this large amount of data in a short amount of time is difficult.
3. Handling heterogeneous data, such as those presented in tweets, is not trivial.
4. Building a system requires expertise from multiple dimensions: data science, big data, software engineering, systems engineering, and natural language processing.
5. Maintaining such systems is expensive for the software and infrastructure required.
6. Updating can be very difficult, depending on the coupling strategy chosen for the modules.

To solve point (1), we can recur to traditional survey methods, such as interviews or surveys. Still, we find them prohibitive, for, besides the high expense, they require a significant amount of time and effort to gather data on a smaller sample of the population. They can only provide information on discrete-time periods rather than a continuous flow. Thus we can look for data in the already public posts on social media. Twitter is a mature, well-established, and popular micro-blogging service that offers users a platform to share their conversations, reviews, and data. For this purpose, we collected a large corpus of heterogeneous COVID-19-related data [1], which we will refer to as *the COVID-19 Twitter chatter data set* and used as our primary source of information. *the COVID-19 Twitter chatter data set* includes raw text, tweet metadata, images, videos, URLs, popularity, and other types of metadata. This corpus is an excellent candidate

for sentiment analysis to follow public opinion on any given topic or event as long as it is related to the COVID-19 pandemic. Still, it does present several challenges, such as the high computing resources needed for conducting the research work, but in return, it provides a curated, well-defined data corpus. In addition, sentiment analysis on a near-real-time basis is possible thanks to the big data technology stack, which is focused on handling and processing a large volume of data at a fast velocity, and from numerous heterogeneous sources [3].

Sentiment analysis refers to a group of natural language processing techniques that allow extracting affective indicators from raw text to determine the sentiment polarity of a given tweet, whether the tweet expresses a positive or negative emotion. To measure the sentiment polarity of tweets, we may employ several language models, each implemented using different technologies and having other characteristics. All implementations use nonlinear statistical models as language representations, in different ways, from massive attention-based deep learning architectures to more straightforward dictionary-based deployments, as is VADER (Valence Aware Dictionary and sEntiment Reasoner) [9].

VADER is an open-source, rule-based tool that recognizes standard terms, idioms, jargon, and more complex grammar structures such as punctuation, negations, abbreviations, etc., commonly employed in social media platforms. VADER uses a curated lexicon of over 7,500 standard terms rated by ten independent humans. VADER has been extensively validated for Twitter-based content, showing promising results in terms of accuracy for tweets in several sentiment analysis tools [2]. However, state-of-the-art language models are implemented by deep neural networks, allowing the use of a more sophisticated text representation space, context awareness, and powerful nonlinear statistical models to provide text classification. BERTweet [20] is based on BERT [4], using a pre-training procedure similar to that utilized by RoBERTa [17] and uses publicly available tweets in English for training and evaluation. TimeLMs [18] introduces a time concept into the language model by utilizing continuous learning and thus accounts for future and out-of-distribution tweets it might encounter.

This language model also uses publicly available tweets in English for training and evaluation. Thus, it is essential to have a system independent of the language model selected for the task, making it simple to choose and change the language model if needed. This work describes the architecture developed to perform this sentiment analysis study, already reported by [16], going in-depth on the design, advantages, and disadvantages of utilizing big data in serverless cloud-based architectures for a project of this size. Next, we present a brief exploration of related works, followed by the methodology followed in the study, the architecture implemented as well as infrastructure-related information, the experiments performed, and closing with a description of the results and a brief discussion of those results.

2 Related Work

Twitter has been widely used to perform sentiment analysis studies in the economic, social, and political domains [27]. Sentiment analysis research extensively

uses Twitter-related traffic, partly due to the high volume [10], high availability, and the limit of 280 characters per entry [8]. A survey shows that building such systems running on private clouds is feasible, relying on the Hadoop tech stack. However, these systems are expensive: they require a significant up-front investment, require effort to set up and maintain and fail to scale properly according to the present demand [5].

Table 1. Summary of massively distributed systems for performing sentiment analysis over large volumes of tweets [16].

Reference	Tech	Batch/Stream	Features	Comments
Victor and Lijo, 2919 [26]	`Hadoop & Spark`	both	HBase interface	
Sathya et al., 2012 [22]	`Hadoop`	batch	classifier selection, pre-processing, sarcasm, VR	open source, needs self hosting
Bhuvaneswari et al., 2019 [2]	`Hadoop & Kafka`	both	Uses flume	
Cenni et al., 2018 [3]	`Hadoop`	batch	-	aggregation of 4 projects
Sehgal and Agarwal, 2016 [23]	`Hadoop`	batch	-	
Kummar and Bala, 2016 [14]	`Mahout`	batch	-	Less complex to build, experiments made on a single node
Kumamoto et al., 2014 [13]		batch	graphs	no details
Khuc et al., 2012 [11]	`Hadoop`	batch	HBase interface	nice UI for data cleanup
Marcus et al., 2011 [19]	`Hadoop`	batch	peak detection, sub-event selection	
this work	`cloud & serverless`	both	auto scaling, easy to consume, cloud native	uses remote hosting

Table 1 displays a summary of these systems, showcasing that Apache Hadoop technologies support most prototypes for streaming and batch data

processing. While these systems all present good results, they also lack flexibility and scalability, resulting in high maintenance costs, and a robust up-front investment is required. Being self-hosted solutions, it is the responsibility of the implementing institution to procure equipment, networking, setup, and maintenance of the system. There is also waste in terms of regular under-utilization of the system, making this an expensive solution that requires a significant up-front investment. So, it can be mitigated by having dynamic scaling in place, thus providing the needed resources instead of having a fixed capacity regardless of utilization.

3 Method

We consumed a large dataset of tweets collected from an open-access repository of global COVID-19-related tweets, called the *COVID-19 Twitter chatter dataset* [1]. It is designed to collect every tweet posted that is somehow related to the pandemic in a diverse variety of geographic locations. This repository provides a list of tweet IDs, geographical location, and detected language using the following schema: `[tweet_id, date, time, lang, country_code]`. However, we encountered schema inconsistencies over time. For example, the annotation of `country_code`, which is necessary for filtering before requesting a tweet lookup, was not introduced until the second half of the year, and even so, a vast number of tweets lack this metadata annotation.

For this reason, we had to load them via Twitter's public API to filter out tweets originating from outside Mexico, which may leave data out from those users who choose not to share their location. We used this information to download each tweet in Mexico, discarding all other metadata provided by Twitter's API for privacy reasons. Specifically, we retrieved COVID-19-related tweets posted in Mexico from February 1, 2020, through December 31, 2020, processing $n = 760,064,879$ unique tweets. All tweets were scrubbed of any personally identifiable information to ensure the user's privacy and comply with ethical, social media use practices, resulting in the following simplified schema: `full_text, id, timestamp`. It is worth mentioning that this data set includes tweets in multiple languages, including both English and Spanish, for many of the population engage in social media in languages other than native Spanish.

Figure 1 summarizes the general data flow of the system. Tweets are consumed parallel to mitigate the official lookup API's rate limitations. The design of this system allows for this API to be easily swapped for other end-points, such as the search end-point, allowing data consumption as a near real-time data stream without the need to perform any other code or system changes. These tweets flow into a pre-processing stage, where they are cleaned up and made ready for consumption by the language models and further narrowed down to a total of over $n = 2,142,800$ unique tweets by discriminating tweets to include only the ones that are not retweets, and, posted from within Mexico. For a detailed definition of the schema returned by the lookup API, please consult the tweet object model definition [24]. Still, for this study, we strip down all

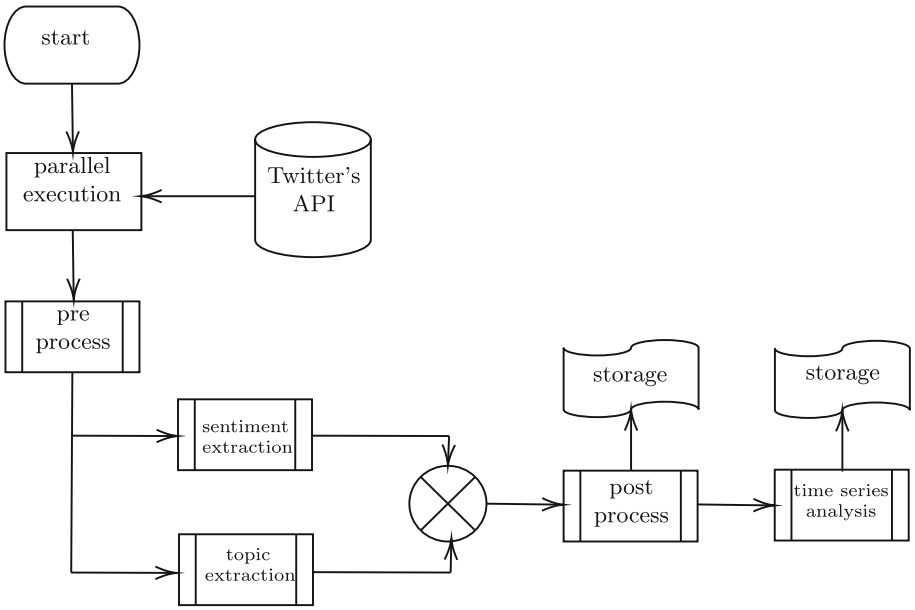

Fig. 1. *Data Flow Overview.* The data is processed in three main stages: first, we load the desired tweet IDs from the *COVID-19 Twitter chatter dataset*, then we consult them directly from Twitter using the official APIs. For pre-processing we clean-up and filter the data, then we process this data set in order to produce a time series of the perceived COVID-19 related sentiment.

the metadata to include only the ID, tweet text, and timestamp. For the pre-processing performed on the text, we follow the standard practices common in natural language processing projects: removing punctuations and emojis, URLs, stop words, converting the text to lower casing, tokenization, stemming, and lemmatization. We then perform sentiment analysis and a time series analysis on the remaining tweets to do the actual sentiment polarity tracking, better described by [16].

Next, a quick summary of the methodology followed by [16] is presented. Natural language processing has several architectures available to implement language models, each with their differences in robustness and accuracy for several tasks, such as sentiment polarity determination. These implementations use non-linear statistical models as language representations, in different ways, from vast attention-based deep learning architectures to more straightforward dictionary-based deployments, such as VADER. VADER [9] is an open-source rule-based robust language model that can handle complex grammar structures commonly employed in social networks. VADER is reliable, fast to deploy, and needs few resources to evaluate new text entries. However, for training, it utilizes a curated corpus evaluated by humans, making adapting it, or incorporating new data a difficult task. BERTweet [20] is based on BERT [4], using a pretraining proce-

dure similar to that utilized by RoBERTa [17] and uses publicly available Tweets in English for training and evaluation.

We then utilize VADER for sentiment polarity determination, consuming a single tweet and providing three different metrics: positive and negative intensity, and a composed metric obtained by normalizing both the positive and negative scores and using an external factor to better approximate a 1 to −1 distribution [9],

$$norm\_score = \frac{sum\_polarities}{\sqrt{sum\_polarities^2 + \alpha}} \tag{1}$$

where *sum_polarities* is the simple addition of positive and negative polarities, and α is initialized as $\alpha = 15$. We need to adjust this α for every operation based on a heuristic and the lexicon collected by the language model. We then use these three metrics to construct a smoothed time series, perform a daily average and calculate an ARIMA model where the box-test showed a $p - value < 2.2e^{-16}$ to estimate the trends and seasonality the series might show. Details on the results obtained are presented in the Results section.

3.1 General System Architecture

Implementing cloud technologies, a serverless architecture, and industry-standard ML-ops practices require three challenges to be properly addressed in order to be successful:

1. The system needs to ingest large amounts of data in the shortest possible time.
2. The system must maintain user privacy and keep the data secure.
3. Use state-of-the-art deep-learning-based language models and implementations with low-level hardware optimization for efficient data processing.

So, the system was implemented using Google's Cloud Platform (GCP)for its dynamic scaling of managed infrastructure and tight integration with the Tensorflow/Keras technology stack, enabling dynamic scaling, loose coupling, and managed micro-services/serverless technology. All while maintaining low-level hardware optimization over the use of multiple CPUs and GPUs in each managed instance.

Figure 2 summarizes the system's general architecture, which follows a flow somewhat similar to that described by Fig. 1. This flow is next described, but keep in mind we have observed some details out of the list, such as monitoring, logging, deployment pipelines, and general error handling.

1. Data is ingested from Twitter, using the identifiers provided by the *COVID-19 Twitter chatter dataset* and the public query API provided by Twitter.
2. The queried tweet is posted in `PubSub`. Then preprocessing is evaluated, writing the resulting clean tweet directly to disk storage and posted again to `PubSub`.

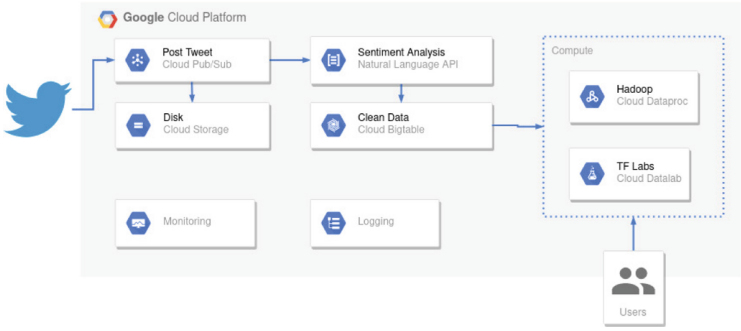

Fig. 2. *General Architecture implemented in Google Cloud Platform (GCP)* [16]. This cloud-based architecture ingests tweets using the official Twitter APIs, sending each one of these through Google's Pub/Sub, which uses as endpoints basic pre-processing, raw storage, and a serverless function to calculate the sentiment polarity. The results of this function is fed into BigTable through another Pub/Sub pipeline.

3. `PubSub` feeds this data entry into a service, which evaluates the polarity of the tweets using a language model implementation written in `TensorFlow`, and posts the results again in `PubSub` to be fed into `BigTable` for final consumption. Note that this evaluation greatly benefits from choosing instances with GPU support, or if hosting the model in another `GCP` service, this will happen automatically to use the `CUDA` technology.
4. The data is now ready for consumption by a managed `Dataproc` instance which can work under different approaches:
   (a) Periodic batch jobs that collect daily aggregations. These aggregations are also stored in cloud storage for easy access.
   (b) `Jupyter` notebooks for manual data exploration.
   (c) A third hook can be placed here for generating near-real-time visualizations of the gathered data.

   Here we decided on the daily aggregations generated by the batch jobs, but this can be easily changed, and the options are not mutually exclusive.

This approach makes it easy to collaborate remotely and share progress or data. All the code was implemented using standard `Python 3.7` and its data-focused libraries. The language models used for polarity calculation were implemented in `TensorFlow`. `TensorFlow` enables consuming state-of-the-art language models as a service, decoupling this architecture from the rest of the solution and allowing the implementation of an automated ML-ops flow to inject updates and model changes. Note the clear separation of operations performed on the ingested data. This flow allows for consuming data as streams, thus allowing using this solution as a decision-making tool by providing near real-time data processing.

Remember that point (3) is the sentiment polarity evaluation triggered by `PubSub` and supports multiple endpoints. This makes it easy to swap solutions

or keep multiple language models evaluating data in parallel and putting results in different `BigTable` instances. We used for sentiment polarity evaluation the `VADER`, `BERTweet` and `RoBERTa` language models, for they cover both state-of-the-art implementations as well as rule-based, well tested systems. Both `BERTweet` and `RoBERTA` are based on `BERT`, which needs around 350M parameters to perform forward and back propagation, needed to evaluate sentiment polarity [4]. However, the system can keep up with tweet consumption by exploiting `CUDA` technology, both in the software and hardware layer, thus providing reliable results promptly.

For point (4), `Dataproc` provides a managed `Apache Hadoop/Spark` cluster that provides Big Data capabilities. We choose to perform data analytics on an extensive batch data set. However, `Apache Spark` also provides streaming capabilities, and this combined with infrastructure-as-code practices, makes trivial the decision of batch vs. streaming data analytics. This flexibility permits batch-processing large amounts of data or providing near real-time data results by processing it as it is consumed in a streaming fashion. Both are proven to be helpful in different use cases.

It is also worth mentioning the MLops practices used throughout this project and not described by Fig. 2. Besides hosting the utilized code in `GitHub`, we integrated and utilized `Google Cloud Deployment Manager` to describe as plain text files the services and infrastructure used, detect changes in them, and automatically propagate changes in the solution. More details on the infrastructure are provided in the next section. This practice, coupled with tagging and versioning of the utilized language models, allows for auto-deployments with a solid integration with `TensorFlow Serving` and infrastructure management as if it were code. While this practice requires robust up-front investment in development time, it provides flexibility and resources to maintain a high-quality standard in the development and experimentation cycle, such as pair programming, code reviews, and automated unit testing.

4 Experiments

This research work consumed a large dataset of tweets, collected from an open-access repository called the *COVID-19 Twitter chatter dataset* [1], and is composed of COVID-19-related tweets, in multiple languages and all over the world. From them, we downloaded $n = 760,064,879$ unique tweets using the official Twitter API, regardless of language but confined to the geographical region of Mexico and from February 1 to December 31, 2020. These were filtered down to $n = 2,142,800$ unique tweets and stripped of any metadata, with light text preprocessing to clean URLs and similar operations. The sentiment polarity was then calculated using several language models, from which a time series was generated, followed by a similar analysis to that of [16].

The system was designed following a serverless, cloud-based architecture as described in Sect. 3, Method. Our system was implemented entirely in the Google Cloud Platform, using `Python 3.7` for any code development, data transformation, or aggregation, as well as language model implementations.

Table 2. Summary of machine types provided by GCP. Note that there are multiple configurations for GPUs as well, but these are only available in A2 instances.

Instance type	CPUs	Memory	GPUs	SSD	interruptable
E2 (grl)	2∼32	0.5∼8 GB	0	no	yes
E2 (shared)	0.25∼1	0.5∼8 GB	0	no	yes
A2	12∼96	7 GB	1∼4	yes	yes
A2 (mega)	96	14 GB	1∼16	yes	yes

However, there is a hard limit on Twitter's consumption rate, set to 450 requests per 15-min window, and a ceiling of total tweets consumed [25]. GCP offers multiple configuration options, considering the number of machines to be provided, although the load-balancer dynamically decides this and the type of machine. Table 2 summarizes the machines types [7], from which we set up three different environments:

1. Reduction costs by selecting the `E2` machine family, which has a low number of shared CPUs and no GPU option.
2. Providing a fast throughput by selecting the `E2` machine family, which comes with a fair number of dedicated CPUs and has multiple GPU options, and finally,
3. A hybrid approach, in which we utilize the data pipelines, but the processing is performed locally.

Table 3. Summary of performance of GPUs offered by GCP, in `TFLOPS`.

Metric	A100	T4	V100	P4	P100
FP64	9.7	0.25	7.8	0.2	4.7
FP32	19.5	8.1	15.7	5.5	9.3
FP16					18.7
INT8				22	
FP64	19.5				
TF32	156				
FP16	312	65	125		
INT8	624	180			
INT4	1248	260			

The local setup consists of a single laptop machine, with 8 CPU cores of Intel's i7-6700 at 2.6 GHz processors, 16 GB, and a 950 m GPU which provides 640 Maxwell GPU cores for a maximum of 1439 `GFLOPS` for FP32 operations and

44.96 GFLOPS for FP64 operations. The GPU options offered by GCP are summarized in Table 3, coming at higher costs and significantly better performance [6]. The language models were implemented using Python 3.7 and TensorFlow with Keras, providing a tight hardware integration, allowing us to take advantage of the GPUs equipped fully. The following section summarizes and compares the results obtained in these configurations.

5 Results

Our system was built to be able to consume a large dataset of tweets collected from an open-access repository of global COVID-19-related tweets, called the *COVID-19 Twitter chatter dataset* [1]. The system was designed to collect every tweet posted related to the pandemic in various geographic locations, with over $n = 760,064,879$ unique tweets, all COVID-19 related and written in multiple languages. These tweets flow into a pre-processing stage, where they are cleaned up and made ready for consumption by the language models and further narrowed down to a total of over $n = 2,142,800$ unique tweets. Then, the algorithm goes into a sentiment analysis phase, where the tweet's sentiment polarity is calculated using multiple language models; in particular, the method utilized to evaluate the consumed tweets used the following implementations: VADER, BERTweet, and RoBERTa. Then, the data is analyzed as a time series, following the same methodology set by [16].

Note that the system could consume and process this large amount of tweets, storing partial results for each stage of the data flow and keeping separated versions running in parallel for the multiple language models utilized. One considerable limitation is Twitter's API rate limit, which is enforced to be 450 requests per 15-minute window, and comes with a low-ceiling cap on the total of tweets consumed. However, a particular academic rate helps these hard limits [25].

To keep costs low, the method utilized the E2 machines types provided by GCP [7] and summarized in Table 2, which comes with a low number of shared CPUs, reaching an evaluation rate of around 4 tweets per second using both BERTweet and RoBERTa. Keep in mind that these models depend on around 350M parameters. Still, this consumption rate is slightly faster than Twitter's rate limit, so this configuration provides a constant throughput while keeping the costs low. Also, note that these rates use low-cost E2 machine types with no GPU options. GPUs can be configured independently of the machine type, as long as it is in the A2 family [6].

A single GPU can provide performance of several TFLOPs, exploited well by the TensorFlow environment. Table 3 summarizes the performance achieved by the multiple GPU family types made available by GCP, all measurements are shown in TFLOPs. However, to keep costs further down, the method exploited the independence of the modules. We evaluated several of the sentiment analysis tasks in a local laptop, with 8 CPU cores of Intel's i7-6700 at 2.6 GHz processors, 16 GB, and a 950m GPU which provides 640 Maxwell GPU cores for a maximum of 1439 GFLOPS for FP32 operations and 44.96 GFLOPS for FP64 operations. While

it is not ideal, the method did see a significant speed-up over the CPU-only evaluation of language models. Thus, choosing the language model based on the target performance and budget is recommended, which can be processed with CPUs and VADER with no issues whatsoever or use a powerful, yet expensive BERT architecture and GPUs.

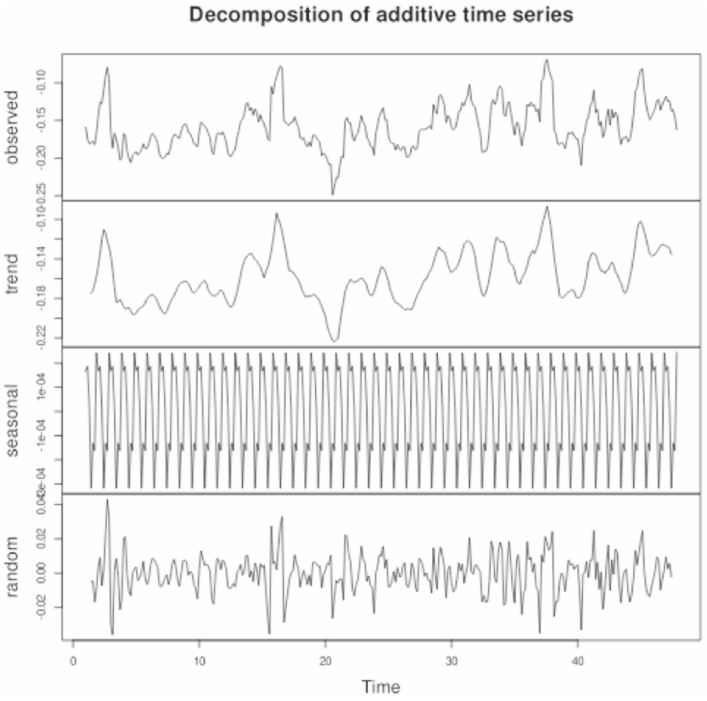

Fig. 3. *Year-long time series of daily averaged of compound sentiment polarity of COVID-19 related tweets, de-trended, in Mexico.* This represents a time series based on the same dataset collected by [1] and restricted to Mexico, from February 2 to December 31 2020. Data was smoothed over via a 7-day rolling means.

Figure 3 shows a summary of the study results. For more details, please refer to [16], but suffice to say that it was possible to perform on time. As can be observed, an almost zero trend is present in the time series, with a slope of $y = -2.2087107971 + 0.0001110643x$ and an ARIMA model fit with a very small p-value (of $2.2e^{-16}$). We also observe a robust weekly seasonality, which is expected from the Twitter data and was also observed by [12]. Another point to note is the change in tweet volume. Before the pandemic declaration, the average volume of tweets in Mexico was 20,971 per day, adding to a total of 608,170 tweets for the month. March presented an average of 46,767 tweets per day and 1,449,768 tweets for the month alone. This represents an increase in the volume of 238.382% between these two months.

6 Conclusions

We presented a flexible system design that supports dynamic scaling and is capable of promptly consuming a large amount of data, which was then used to track the emotional well-being of the Mexican population from February to December 2020. For this, we consumed more than $n = 760,064,879$ unique tweets, all related to COVID-19 and written in multiple languages. We have shown that this system can process such a large amount of data, both in batch and streaming modes, while still being capable of swapping out multiple language models and facilitating data exploration and visualization. For the language models, we worked with both rule-based models, VADER, and state-of-the-art attention-based deep learning models, with BERTweet and RoBERTa, by simply plugging them into the appropriate stage in the data pipeline. It combined with modern MLops practices such as auto-deployments, model versioning, and infrastructure-as-code allows for a cost-effective solution for this big data problem. It mitigates common issues such as model performance degradation and adversary attacks and can gracefully handle data volume changes that occur naturally.

We have shown that it is possible to build a large-scale big data system for sentiment analysis using serverless technology and still achieve high performance. So near-real-time results by taking advantage of low-level code and hardware optimization. In our case by exploiting `Tensorflow` and its `CUDA` integration. This architecture presents additional advantages over traditional extensive data systems. For example, no significant initial investment is required, the software overhead and maintenance costs are significantly lower than those of `Apache's Hadoop`, and dynamic scaling is possible without any code changes. However, this system does present some drawbacks. The hosted services are off-site, with little to no control on our part, making security concern of the highest importance. To mitigate this concern, we have stripped the data of any metadata that could be used to identify a particular individual, such as account IDs. It's worth noting that the data utilized, while being a large corpus, is limited to a single year and country, that of 2020 in Mexico. There is no technical reason for having this limitation, and the system is fully capable of consuming an enormous amount of data. Also, in the future, we would like to develop visualization dashboards to provide instant feedback to decision-making organizations and to see a more extensive adoption in many organizations that include emotional data in their decision-making policies.

References

1. Banda, J.M., et al.: A large-scale COVID-19 twitter chatter dataset for open scientific research - an international collaboration, February 2021. https://doi.org/10.5281/zenodo.4540809
2. Bhuvaneswari, M., et al.: Handling of voluminous tweets and analyzing the sentiment of tweets. In: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 360–364. IEEE (2019)

3. Cenni, D., Nesi, P., Pantaleo, G., Zaza, I.: Twitter vigilance: a multi-user platform for cross-domain twitter data analytics, NLP and sentiment analysis. In: 2017 IEEE SmartWorld Ubiquitous Intelligence and Computing, Advanced and Trusted Computed, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation, SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2017, pp. 1–8 (2018). https://doi.org/10.1109/UIC-ATC.2017.8397589
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv: Computation and Language (2018). MAG ID: 2896457183
5. El Alaoui, I., Gahi, Y., Messoussi, R.: Full consideration of big data characteristics in sentiment analysis context. In: 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp. 126–130. IEEE (2019)
6. Google: GCP's GPU offering (2022). https://cloud.google.com/compute/docs/gpus
7. Google: GCP's machine types (2022). https://cloud.google.com/compute/docs/machine-types
8. Heisler, Y.: Twitter's 280 character limit increased engagement without increasing the average tweet length (2018). https://bgr.com/2018/02/08/twitter-character-limit-280-vs-140-user-engagement/
9. Hutto, C., Gilbert, E.: Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 8 (2014)
10. InternetLiveStats.com: Internet live stats (2019). https://www.internetlivestats.com/one-second/#tweets-band
11. Khuc, V.N., Shivade, C., Ramnath, R., Ramanathan, J.: Towards building large-scale distributed systems for twitter sentiment analysis. In: Proceedings of the ACM Symposium on Applied Computing, pp. 459–464 (2012). https://doi.org/10.1145/2245276.2245364
12. Kmetty, Z., Bokányi, E., Bozsonyi, K.: Seasonality pattern of suicides in the us-a comparative analysis of a twitter based bad-mood index and committed suicides. Intersect. East Eur. J. Soc. Polit. **3**(1), 56–75 (2017)
13. Kumamoto, T., Wada, H., Suzuki, T.: Visualizing temporal changes in impressions from tweets. In: ACM International Conference Proceeding Series, pp. 116–125, 04–06 December 2014. https://doi.org/10.1145/2684200.2684279
14. Kumar, M., Bala, A.: Analyzing twitter sentiments through big data. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 2628–2631. IEEE (2016)
15. Laney, D., et al.: 3D data management: controlling data volume, velocity and variety. META Group Res. Note **6**(70), 1 (2001)
16. León-Sandoval, E., Zareei, M., Barbosa-Santillán, L.I., Falcón Morales, L.E., Pareja Lora, A., Ochoa Ruiz, G.: Monitoring the emotional response to the COVID-19 pandemic using sentiment analysis: a case study in Mexico. Comput. Intell. Neurosci. **2022** (2022). https://doi.org/10.1155/2022/4914665. publisher: Hindawi
17. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv: Computation and Language (2019). MAG ID: 2965373594
18. Loureiro, D., Barbieri, F., Neves, L., Anke, L.E., Camacho-Collados, J.: TimeLMs: diachronic language models from twitter. arXiv:2202.03829 (2022), http://arxiv.org/abs/2202.03829

19. Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: Twitinfo: aggregating and visualizing microblogs for event exploration. In: Conference on Human Factors in Computing Systems - Proceedings, pp. 227–236 (2011). https://doi.org/10.1145/1978942.1978975
20. Nguyen, D.Q., Vu, T., Nguyen, A.T.: BERTweet: a pre-trained language model for English tweets. arXiv:2005.10200 (2020), http://arxiv.org/abs/2005.10200
21. Ramírez, H.L.G.: Twitter - undersecretaries of prevention and health promotion (2020). https://twitter.com/HLGatell/status/1233245568668966913
22. Sathya, V., Venkataramanan, A., Tiwari, A., Dev Daksan, P.S.: Ascertaining public opinion through sentiment analysis. In: Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019 (ICCMC), pp. 1139–1143 (2019). https://doi.org/10.1109/ICCMC.2019.8819738
23. Sehgal, D., Agarwal, A.K.: Sentiment analysis of big data applications using twitter data with the help of HADOOP framework. In: 2016 International Conference System Modeling & Advancement in Research Trends (SMART), pp. 251–255. IEEE (2016)
24. Twitter: Twitter's object model definition (2022). https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model/tweet
25. Twitter: Twitter's public API access level policy (2022). https://developer.twitter.com/en/docs/twitter-api/getting-started/about-Twitter-api#v2-access-level
26. Victor, P., Lijo, V.: A big data processing framework for polarity detection in social network data. In: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 291–295. IEEE (2019)
27. Wang, H., Can, D., Kazemzadeh, A., Bar, F., Narayanan, S.: A system for real-time twitter sentiment analysis of 2012 US presidential election cycle. In: Proceedings of the ACL 2012 System Demonstrations, ACL 2012, pp. 115–120. Association for Computational Linguistics (2012)
28. Ylijoki, O., Porras, J.: Perspectives to definition of big data: a mapping study and discussion. J. Innov. Manage. **4**, 69–91 (2016)