informatics

MDPI

Article

Machine Learning and Deep Learning Sentiment Analysis Models: Case Study on the SENT-COVID Corpus of Tweets in Mexican Spanish

Helena Gomez-Adorno [1,*], Gemma Bel-Enguix [2], Gerardo Sierra [2], Juan-Carlos Barajas [3] and William Álvarez [1]

[1] Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Mexico City 04510, Mexico; wikkilicht@ciencias.unam.mx
[2] Instituto de Ingeniería, Universidad Nacional Autónoma de México, Mexico City 04510, Mexico; gbele@iingen.unam.mx (G.B.-E.); gsierram@iingen.unam.mx (G.S.)
[3] Facultad de Ciencias, Universidad Nacional Autónoma de México, Mexico City 04510, Mexico; cbarajas5@ciencias.unam.mx
* Correspondence: helena.gomez@iimas.unam.mx

Abstract: This article presents a comprehensive evaluation of traditional machine learning and deep learning models in analyzing sentiment trends within the SENT-COVID Twitter corpus, curated during the COVID-19 pandemic. The corpus, filtered by COVID-19 related keywords and manually annotated for polarity, is a pivotal resource for conducting sentiment analysis experiments. Our study investigates various approaches, including classic vector-based systems such as word2vec, doc2vec, and diverse phrase modeling techniques, alongside Spanish pre-trained BERT models. We assess the performance of readily available sentiment analysis libraries for Python users, including TextBlob, VADER, and Pysentimiento. Additionally, we implement and evaluate traditional classification algorithms such as Logistic Regression, Naive Bayes, Support Vector Machines, and simple neural networks like Multilayer Perceptron. Throughout the research, we explore different dimensionality reduction techniques. This methodology enables a precise comparison among classification methods, with BETO-uncased achieving the highest accuracy of 0.73 on the test set. Our findings underscore the efficacy and applicability of traditional machine learning and deep learning models in analyzing sentiment trends within the context of low-resource Spanish language scenarios and emerging topics like COVID-19.

Keywords: sentiment analysis; COVID-19; machine learning; social media; Spanish

Citation: Gomez-Adorno, H.; Bel-Enguix, G.; Sierra, G.; Barajas, J.-C.; Álvarez, W. Machine Learning and Deep Learning Sentiment Analysis Models: Case Study on the SENT-COVID Corpus of Tweets in Mexican Spanish. *Informatics* **2024**, *11*, 24. https://doi.org/10.3390/informatics11020024

Academic Editor: Olga Kurasova

Received: 22 February 2024
Revised: 23 March 2024
Accepted: 16 April 2024
Published: 23 April 2024

1. Introduction

Social media communication is crucial in all sectors of the population's life. Companies use social media to massively promote products and services, while people use them to transmit experiences and opinions. Natural Language Processing (NLP) and Text Mining have been of great interest in exploring this source of textual communication to generate information about mass behavior, thoughts, and emotions on a wide variety of topics, such as product reviews [1], political trends [2], and stock market sentiment [3]. During the Coronavirus pandemic, people expressed how they experienced the consequences of quarantine, the way it altered the daily rhythm of life, and how they changed their day-to-day activities.

Among the most used social media during the pandemic was Twitter, which at the time functioned as a freely accessible universal microexpression tool. This made it an ideal platform to capture the population's feelings during this historic moment. Many studies have been presented that analyze various aspects of the epidemic, some of them on Twitter and mainly in English.

This article presents the work carried out to study the emotional impact of COVID-19 on the Mexican population. The MIOPERS platform responded to UNAM's initiative to develop models for the analysis and visualization of information that support strategic decision-making, especially during lockdown. During the pandemic, there were two main motivations for starting such work: (a) to evaluate people's behavior, moods, and popularity of the measures given by the government and (b) to monitor users with possible symptoms.

This initiative, which covers two years (2020–2022), the duration of the pandemic, allowed a compilation of many tweets related to COVID-19. This facilitated the study of topic-related lexicon, mentions, and hashtags, which in turn served as a basis for studying other important NLP topics, such as sentiment analysis.

This article focuses on developing a specific corpus for polarity analysis of COVID-19, the SENT-COVID corpus, taking a subset of the tweets collected by the Miopers system during the pandemic. Furthermore, polarity classification experiments are performed, applying both traditional ML and DL methods. To do this, the article follows the structure explained below. Related work is discussed in Section 2, especially on sentiment analysis in social networks or specifically oriented to the topic of COVID-19. Section 3 explains the compilation of the corpus, the annotation protocol, and the agreement results. The methodology that has been followed to carry out the analysis is described in Section 4, including pre-processing, forms of text representation, and algorithms used. The results are presented and discussed in Section 5. The article concludes with the conclusions in Section 6.

2. Related Work

Numerous toolkits are available to process textual data, which makes complex NLP tasks more accessible with user-friendly interfaces. In the context of sentiment analysis, several researchers have used libraries such as TextBlob, VADER, and Pysentimiento, among others. TextBlob and VADER have the advantage of not requiring training data, as it is a lexicon-based approach. Therefore, they have been popular tools for analyzing comments on social networks, such as tweets [4–8], youtube [9–13] or Reddit [14–17] comments. Although the lexicon-based approach is suitable for general use, its main limitation lies in its difficulty adapting to changing contexts and linguistic uses [18]. Examples are texts such as tweets that have a lively and casual tone [19]. In addition, if we look at those related to COVID-19, we find new terms associated with the phenomenon. Additionally, since TextBlob and VADER were designed mainly for English-language texts, they may not be as effective when used in texts in other languages. Therefore, a toolkit for analyzing text sentiments and emotions in a wide range of languages is the Pysentimiento library, which offers support for multiple languages [20–22], including Spanish [23,24]. Furthermore, Pysentimiento uses state-of-the-art machine learning models, such as BERT (Bidirectional Encoder Representations from Transformers) models, for sentiment analysis. However, this requires more computing resources than TextBlob or VADER.

From the beginning of the quarantine period, several researchers studied social media information to measure people's feelings about their situation during the COVID-19 pandemic [25]. This has been done considering the language and domain of the comments posted on the different social platforms [26]. Many studies have used TextBlob, VADER, and Pysentimiento tools for sentiment analysis on social networks [6,23,27–31]. Moreover, machine learning approaches have been widely adopted to categorize sentiments into two (negative and positive) or three classes (positive, negative, and neutral). For example, Long Short Term Memory (LSTM) recurrent neural network has been used in Reddit comments, which allows for 81.15% accuracy [32].

Chunduri and Perera [33] have used advanced deep learning models, such as Spiking Neural Networks (SNN), for polarity-based classification. SNNs encompass what is known as brain-based computing, and attempt to mimic the distinctive functionalities of the human brain in terms of energy efficiency, computational power, and robust learning. Although

they report 100% accuracy with their model, their main claim is that SNNs have lower energy consumption than ANNs.

For public tweets related to COVID-19, the TClustVID model [34] was developed, achieving a high accuracy of 98.3%.

Researchers have also analyzed the performance of language models for sentiment analysis in Spanish. Specifically, for the COVID-19 tweet polarity, Contreras et al. [35] found that pre-trained BERT models in Spanish (BETO), with domain-adjusted, have achieved a high accuracy of 97% in training and 81% in testing. Such performance was the best compared to multilingual BERT models and other classification methods such as Decision Trees, Support Vector Machines, Naive Bayes, and Logistic Regression.

Research has focused not only on creating computational models for text classification but also on annotated datasets, which help to train and evaluate models in supervised learning approaches. An example is COVIDSENTI [36], which consists of 90,000 COVID-19-related English-language tweets collected in the early stage of the pandemic from February to March 2020. Each tweet has been labeled as positive, negative, or neutral. Furthermore, state-of-the-art BERT models have been applied to the data to obtain a high precision of 98.3%.

For sentiment analysis, several corpora of annotated tweets related to COVID-19, mainly in English, have been released [36–40]. However, since the behavior of social media users also varies with language [41], having datasets in various languages besides English is crucial. Therefore, efforts have been made to compile multilingual corpora [42,43] as well as language-specific datasets such as Portuguese [44,45], Arabic [46,47], French [48], among others [49–51]. For the Spanish language, there are annotated tweet datasets for tasks such as hate speech detection [52], aggression detection [53], LGBT-phobia detection [54], and automatic stance detection [55], among others. However, to our knowledge, there is no manually annotated public corpus for the sentiment polarity of COVID-19-related tweets in Spanish. Given that research tends to use an automatic labeling process. Like the work by Contreras mentioned above [35]. Therefore, we present a corpus with a manual labeling process and an annotation guideline. Furthermore, we provided an extensive analysis of the agreement between the annotators.

3. Data Collection and Annotation: The SENT-COVID Corpus

We collected COVID-19 tweets by implementing the Twitter API in Python. The messages are from 1 April 2020, to the end of 2022. About 4,000,000 tweets were collected, including only messages labeled as written in Mexican Spanish. We also included tweets that were responses or retweets, i.e., the type and form of the tweet did not matter to the extraction and annotation process.

Once the data was obtained, we filtered the messages with a dictionary of appropriate terms, hashtags, and mentions depending on the development of the pandemic. In the first lexicon, the terms focused on different variants of the word COVID-19 (*coronavirus, el virus, covid, lo del contagio*) and symptoms. (*dolor de cabeza agudo, cuerpo cortado, diarrea, fiebre [leve], tos [seca], dolor de garganta, altas temperaturas*). Regarding hastaghs, many of them were government messages or slogans, used to support their policies, such as *#QuedateEnCasa, #TecuidasTúNosCuidamosTodos, #SusanaDistancia*. This is shown in Table 1.

After applying the initial filter, our corpus remained at only 4986 tweets. However, we removed 120 tweets that were not in Spanish and those that contained less than three words because they did not provide enough information to assign them any label. Therefore, our final corpus consists of 4799 tweets.

Table 1. Lexicon used to filter the COVID-19 related tweets for the corpus creation.

VARIANTS COVID	SYMPTOMS
COVID-19	me dio diarrea
coronavirus	dolor de cabeza agudo
Covid-19	cuerpo cortado
Coronavirus	fiebre (leve)
Covid19	tos (seca)
Covid	dolor de garganta
lo del contagio	altas tamperaturas
esta pandemia	
Corona Virus	
el virus	
HASHTAGS	**HASHTAGS**
#AbrahamSealaverga	#Covid19
#AburridoEnCasa	#covidmexico
#AislamientoSocial	#CuarentenaCoronavirus
#BastaDeFakeNews	#CuidaALosTuyos
#carroñavirus	#CuidemosALosMayoresYPequeños
#CODVID19	#EnCuarentena
#ConferenciaCovid19	#MeQuedoEnHome
#ConLaFuerzaDeLosProtocolosSI	#NoSonVacaciones
#Coronavirus	#QuedateEnCasa
#CoronavirusMx	#QuédateEnTuCasa
#coronaviruspeleishon	#COVID19mexico
#COVID19mx	#CuandoEstoSeAcabe
#Cuarentena	#cuarentenamexico
#CuidaALosDemas	#CuidarnosEsTareaDeTodos
#Cuidate	#CulturaEnCasa
#encasa	#Enfermera
#MeQuedoEnCasa	#México
#NeumoniaAtipica	#QuedarseEnCasa
#quédate	#QuédateEnCasaUnMesMas
#QuedateEnLaCasa	#QuedateEnTuCasaCarajo
#quedateentuputacasaalaverga	#QuedenseEnCasa
#QuePorMiNoQuede	#sabadodecuarentena
#SaltilloQuédateEnCasa	#SeFuerteMexico
#SiTeSalesTeMueres	#StayAtHome
#StayAtHomeAndStaySafe	#Super
#SusanaDistancia	#teamwork
#TecuidasTúNosCuidamosTodos	#TipsDeCuarentena
#ÚltimaHora	#UltimaOportunidad
#YaBastaDeFakeNews	#yolecreoagattel
#YoMeQuedoEnCASA	

3.1. Annotation Protocol

We created an annotation guideline, summarized in this section, based on the polarity of sentiments. This describes how we labeled tweets and the criteria we used to categorize sentiments into three classes: positive, negative, and neutral. Each tweet in the corpus was manually assigned to one of the three categories.

We used as reference the Robert Plutchik's description of the eight primary emotions [56]-anger, fear, sadness, disgust, surprise, anticipation, trust, and joy. This allowed us to describe the polarity categories as follows.

POSITIVE TAGS. Positive tags are used to identify tweets that communicate joy/trust. Positive tweets are characterized by:

1. Predominance of pleasure or well-being:

- 'No se ustedes pero yo he sido muy feliz durante esta cuarentena'.
  [I don't know about you but I have been very happy during this quarantine].
- 'jaja Ana me acaba de alegrar la cuarentena'.
  [haha Ana just made my quarantine happy].

2. Cultivation of personal strengths and virtues that lead to happiness.
   - 'Desde que inició la cuarentena le ando dando duro al ejercicio y a la dieta'.
     [Since the quarantine began, I have been doing exercise and diet.]
   - 'Algunos están estresados, preocupado yo digo #GraciasCuarentena porq me ha hecho valorar tanto, porque la salud es primero y solo se valora cuando se pierde'.
     [Some are stressed or worried but I say #ThanksQuarantine cause it has made me appreciate a lot of things, since health comes first and health is only appreciated when lost.]
3. Optimization of health, psychological resilience and promotion of efficient, flexible and creative reasoning.
   - 'Si, en una época de crisis económica, saqué adelante mi economía con una olla de tamales y no me avergüenzo. Fui muy feliz; trabaje duro; hice muchos amigos y sobre todo aprendí a ser humilde'.
     [Yes, in a time of economic crisis, I managed my economy with a pot of tamales and I am not ashamed. I was very happy; I worked hard; I made many friends and above all I learned to be humble.]
   - 'Con esta cuarentena aprendí apreciar el hoy, reírse de uno mismo, valorar a los que están, animarse, dar amor porque si confiar porque si, y perdonar porque si. Estos días me motivaron a . . . '.
     [With this quarantine I learned to appreciate the present, to laugh at myself, to value those who are with me, to cheer up, to give love just because, trust just because and forgive just because. These days motivated me to . . .]
4. Motivation to achieve the life goals people set for themselves.
   - 'Compré un vestido para una boda en la playa, sin invitación a ninguna boda en la playa y en plena cuarentena. Me gusta soñar'.
     [I bought a dress for a beach wedding, with no invitation to any beach wedding and in the middle of quarantine. I like to dream.]

NEGATIVE TAGS. Negative tags are used to identify tweets that communicate anger, fear, or sadness. Negative tweets are characterized by:

1. Expressing situations where there is something unpleasant or violent
   - 'Oigan, me duele el pecho medio raro ¿Es síntoma de covid o ya me volví loca?'
     [Hey, my chest hurts kind of weird. Is it a symptom of covid or have I gone crazy?]
2. Representing a barrier to achieving a goal or requires the mobilization of resources for the creation and elaboration of plans to resolve a situation
   - 'El número de contagios de COVID-19 en México puede ser hasta 50 veces más que los reportados: Julio Frenk'.
     [The number of COVID-19 infections in Mexico might be even 50 times more than the number reported: Julio Frenk]
   - 'La contingencia continua y las necesidades son cada vez en más familias'.
     [The contingency continues and the needs are increasing in more families]

NEUTRAL TAGS. Neutral cues are used to identify tweets that do not communicate any emotion/sentiment. Neutral tweets are characterized by:

1. Not communicating a specific message.
   - '@ExpansionMx El virus no tiene nada que ver. Lo que sí, es el Gobierno!' [The virus has nothing to do. But the government has something to do with it!]
2. Expressing some kind of doubt, without attacking something or someone.
   - 'No se supone que suspendieron parquímetros por contingencia? @ecoParq @Claudiashein'. [Aren't parking meters supposed to be suspended due to the contingency?]
   - 'Mateo cuando escucho la pregunta de la vacuna del jabón dijo—pero como de jabón? Que no sabe que en las vacunas van la mitad de los virus??' [When Mateo heard the question about the soap vaccine, said—but how about soap? Who doesn't know that half of the viruses are in vaccines??]

To annotate the corpus, we had a previous step. We designed an experiment to check which is the best way to proceed to categorize the messages in type of corpus. Two students were asked to label a sample of 100 tweets with the tags positive, negative, or neutral, without any guidance but based solely on their own opinions.

At the same time, two more students were in charge of labeling the same messages following the guide that had been developed. The analysis of the results showed that the guide favors the agreement between the annotators. Thus, we moved on to a second phase with new students, with the help of the guide. Therefore, the final process of tagging involved three annotators who labeled the tweets according to our created guide.

3.2. Data Statement/Annotators Data

We followed the guidelines specified by [57] to create this data statement.

A. Curation Rationale: We collected tweets from the widely used social media platform, Twitter, due to its convenience in acquiring concise statements from the general user population on diverse topics within a digital context. We used specific key terms and hashtags commonly used to refer to the pandemic.
B. Language variety: We systematically extract a set of tweets by filtering for specific keywords and ensuring that they are in Spanish and geographically associated with the designated region (Mexico).
C. Tweet author demographic: The data is likely to come from a wide range of users with different characteristics such as age, gender, nationality, race, socioeconomic status and educational backgrounds. This is because we collected the data using Twitter's data collection API, which is expected to have a diverse user base in Mexico.
D. Annotator demographic: We selected three annotators from the UNAM Language Engineering Group to label the tweets. All of them were undergraduate students from this university, between 20 and 25 years old, Spanish native speakers with Mexican nationality and residence.
E. Speech Situation: All tweets are about the pandemic. The years of extraction are 2020 to 2022.
F. Text characteristics: The tweets collected come from a pandemic context, so they followed a specific global trend. They could be a unique tweet or a response to another tweet. The limited length of tweets is an important factor to consider, as is the social media policy. All the data are public.
G. Recording Quality: We extracted the tweets from the Twitter API.
H. Ethical Statements: We collected all tweets for academic use according to Twitter's privacy policy.

3.3. Results of the Annotation Process

The "interrater reliability" [58] is a measurement of the extent to which data annotators (raters) assign the same score or label to the same variable. Frequently, this quantity is calculated by Cohen's kappa coefficient (κ). We assessed the agreement of the corpus

annotators by using the Inter-Annotator Agreement (IAA) score, where Cohen's Kappa statistical measure is used in its definition. So, the annotators who did not use the guide presented a Cohen's (κ) score of 0.178, a very slight agreement. In contrast, the annotators that used the guide presented a Cohen's (κ) score of 0.4369, a moderate agreement indeed. Table 2 below shows the percent of agreement and Cohen's (κ) score for each pair of annotators who did not use the guide. Compared to this, Table 3 below shows the percent of agreement and Cohen's (κ) score for each pair of annotators who used the guide.

Table 2. Agreement score by the annotators of the classification of sentiments without a guide.

Annotator Pair	**A&B**
Percent of agreement	41%
Cohen's κ score	0.1785

Table 3. Agreement scores by each pair of annotators of the classification with the guide.

Annotator Pair	**1&2**	**2&3**	**1&3**
Percent of agreement	61%	70%	62%
Cohen's κ score	0.3945	0.5547	0.3716

Cohen's Kappa suits very well for estimating the agreement between not more than two annotators. So, given the characteristics of our annotation process, where we have at least three annotators for each tweet, we used Fleiss' kappa to measure the agreement between the three annotators that used a guide. This analysis resulted in an overall agreement of 0.4369, reflecting a moderate inter-annotator agreement.

Having three annotators for each tweet allowed us to identify the labels with the majority vote. By having three independent labels where there was disagreement, we could seek agreement on two out of three to set the repeated label as the definitive label in the final corpus. At the end of the annotation process, some tweets did not have an assigned label since the three annotators disagreed. For this reason, it was necessary that all the annotators together decide on the final label for the tweets without agreement.

Our final corpus consists of 4799 tweets, of which 1834 (38.21%) contain negative sentiments, 1126 (23.46%) contain positive sentiments, and 1839 (38.33%) contain neutral sentiments. Finally, Table 4 presented below shows the general statistics computed from word counts on each tweet of our corpus. The minimum number of words across all categories is three. We can ascertain the range of words in each category by computing the maximum. Tweets with a negative sentiment have the highest maximum number of words, while those with a positive sentiment have a lower range. The maximum count of words varies significantly between categories. However, the average number of words is quite similar. Additionally, on average, tweets contain a low number of words, which could explain why the standard deviation of the count is so high.

Table 4. General statistics computed from word counts on each tweet.

	Positive Tag	**Negative Tag**	**Neutral Tag**
Average number of words per tweet	22.85	26.39	20.97
Standard Deviation	12.69	15.46	13.59
Variance	161.14	239.02	184.65
Minimum number of words in a tweet	3	3	3
Maximum number of words in a tweet	59	339	88
Total number of words	25,729	48,398	38,580
Tweets count	1126	1834	1840

4. Sentiment Analysis Methods

Once the corpus has been annotated, the next step is to process the raw tweets into data that we can use for classification. This section outlines the sentiment analysis methods we evaluate to build a classification model on COVID-19-related tweets.

Figure 1 shows the workflow of our experimentation, starting with the preprocessing of the text data before feature extraction, which includes removing digits, separating words based on patterns, normalizing words, wrapping special tokens, transcribing emojis, lemmatizing, and removing stop words. The results of processing the tweets using these methods are presented, demonstrating the transformation from raw tweets to processed text. The feature extraction techniques we evaluated are the Bag of Words (BoW) model, Term Frequency-Inverse Document Frequency (Tf-Idf), word embedding, and phrase modeling. Once the features were extracted using the BOW or *n*-grams models, we used feature selection techniques (explained in Sections 5.1 and 5.2) to reduce the vector dimensions while keeping the highest amount of information as possible. The models and algorithms for text classification we evaluated include ready-to-use libraries such as TextBlob, VADER, and the Pysentimiento Toolkit. With respect to the supervised learning models, we evaluated Logistic Regression, Naive Bayes Classification, Support Vector Machines, and Multilayer Perceptron (MLP). Moreover, transformer networks are also explored. Finally, for evaluating the sentiment classification models, we used cross-validation for the traditional machine learning algorithms that required training; for the ready-to-use libraries, we used the entire datasets because no training is required; for the transformers' models, we only did one training test split due to the computational cost of the experiments.

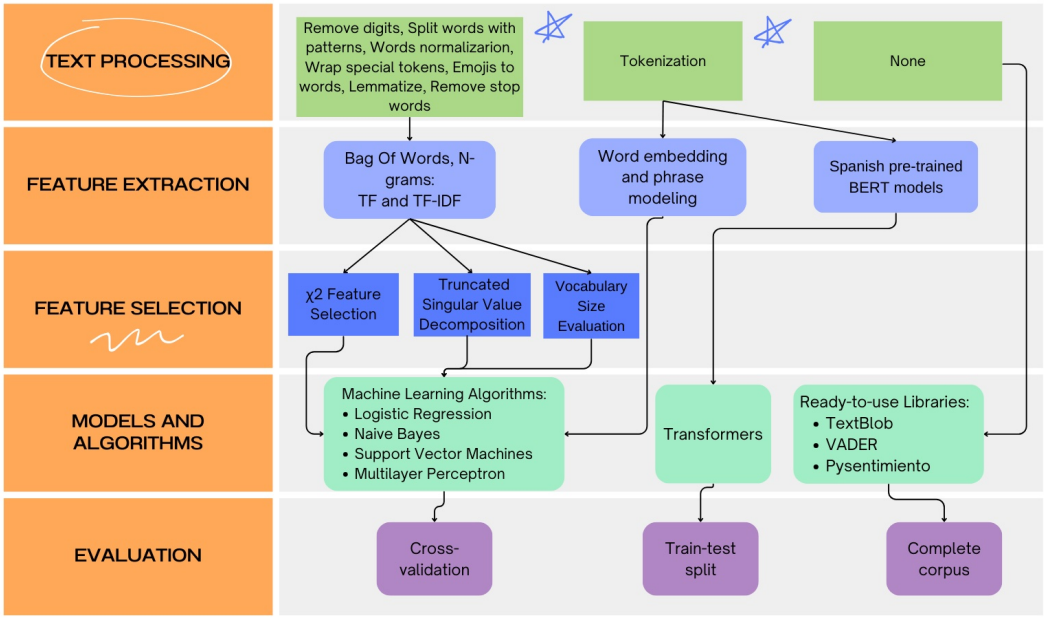


Figure 1. Sentiment analysis experimentation workflow.

4.1. Text Processing

To build a Bag-of-Words (BoW) representation (we consider bi-grams and other structures in Section 4.2) we compose a vocabulary of all unique words in the corpus following the process outlined below:

1. Remove digits, double blanks, and line breaks: Extracting information from digits presents a challenge since they can represent different things such as magnitudes, time-dates, directions, etc. Consequently, digits are intentionally omitted.
2. Separate words with patterns: Tweets contain misspellings or camel case writing in hashtags. So, we identified common patterns (such as dots before a capital letter or symbols like '#' or '¿') and separated them using regular expressions.

3. Normalize words: We transformed text to lowercase and removed punctuation marks and other special characters. It would not have split properly if we had done this before the previous step. In addition, consecutive repeated characters (usually used for laughs) were minimized to two repetitions to prevent the formation of new tokens for words already present in the vocabulary.
4. Wrap special tokens: Tweets often contain mentions, web links, and pictures. Thus, we identified common objects with general labels. For example, mentions to users were wrapped with the token 'usuario' and web links by the token 'url'.
5. Transcribe emojis to words: We convert emojis into words, positioning them properly within the tweet using the emoji (https://pypi.org/project/emoji/ accessed on 20 February 2024) Python module. All of these are normalized to lowercase.
6. Lemmatize: We find a word's dictionary form (or lemma). This process was done using the Spacy library. This allowed us to reduce vocabulary size by avoiding multiple tokens for different inflections of the same word.
7. Remove Stop words: Common words that do not carry semantic information, called 'stop words', are removed to reduce the vocabulary size.

In the following, we show the results of the comments processing using the described method. Table 5 shows the original raw tweet in the first column and the processed text in the second column. After the process, the entire text is in lowercase. The links are changed to the token 'url', while the users mentioned are replaced with 'usuarios'. Furthermore, emojis are converted into words and any repetition is avoided.

Table 5. Original (raw) and processed version of a sample of tweets.

Raw Tweet	Processed Tweet
#CuarentenaNacional #CDMX consulta:https://t.co/TcjustEg	cuarentena nacional cdmx consulta url
Buen díaa!!!#ConCaféEnMano para alegrar la mañana #EnCasa	buen dia con cafe en mano para alegrar la mañana en casa
@CONANPmx@GobiernoMX lleno de gente en Av. Tenorio	usuario lleno de gente en av tenorio
Marcarle a mi preciosita en momento de crisis. 🥺 🥺 🥺	marcarle preciosa momento crisis cara por favor
Uff 😢 🦠 🦠 🦠 #QuedateEnCasa 🏠 #Coahuila #Mexico 😥	uf cara triste alivio microbio quedar casa jardin coahuila mexico
#SNTEsalud ⚕️ ⚠️ ALERTA alto contagio en los mochis	sntesalud simbolo medicina advertencia alerta alto contagiar mochis

4.2. Feature Extraction

After processing the text, we investigated and experimented with different numerical data representations in the training stage. We compared some simple feature extraction techniques and algorithms to build a vocabulary from all words in the corpus. Also, we tried out some pre-trained embedding models, which provide a more complex yet more effective way of extracting text features.

4.2.1. Bag of Words Model

In the BoW model, each token in the text corresponds to a given dimension (feature) in a vector representation. Each token in a text will have a weight that can be given by the number of occurrences of a word in the text (term frequency) or by multiplying the number of occurrences of a word in the entire corpus with the occurrences of a word in the text (Tf-Idf).

1. Term Frequency: To transform tweets to vector representations, we used the Scikit-learn library (https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html?highlight=countvectorizer#sklearn.feature_extraction.text.CountVectorizer (accessed on 20 February 2024)). The size of the vector of

each text is equal to the size of the vocabulary of the corpus, and, as mentioned before, the value of each dimension is the number of times that the words appear in the tweet. We extracted different feature sets by modifying the following parameters:

- n_gram_range: It allows us to know if each token is formed by singular words or n_grams (*n*-gram is a continuous sequence of n items from a given sequence of text) of words (this tries to preserve local ordering of words but at the cost of highly increasing the number of features).
- min_df: It is the minimum number of times that a word must appear in all documents to be considered part of the vocabulary. Many words are present in the corpus with few appearances, so small variations of this parameter highly increase vocabulary size.
- stop_words: We used the stopword list provided by the Nltk library, which consists of 313 words.

Table 6 shows in the last column the vocabulary size extracted by modifying the parameters mentioned above. The first column indicates whether the text is lemmatized or not, the second column indicates whether stopwords are removed, the third column indicates the n_gram range, and the fourth column shows the value of the minimum document frequency.

Table 6. Vocabulary size by different parameter settings.

Text	Stopwords	*n*-gram Range	Mindf	Vocabulary Size
Normalized	Yes	1_gram	2	4343
			3	2797
		2_gram	2	10,674
			3	5919
	No	1_gram	2	4198
			3	2664
		2_gram	2	6456
			3	3614
Lemmatized	Yes	1_gram	2	3663
			3	2422
		2_gram	2	10,213
			3	5812
	No	1_gram	2	3605
			3	2369
		2_gram	2	6533
			3	3629

2. Tf-Idf: The Term frequency weighting treats all words as having the same importance. This can be improved by considering the significance of each term in the corpus. To give each term a weight. A high Tf-Idf score indicates that a word is present in the tweet but not in many other tweets in the corpus. However, a low Tf-Idf value implies that most tweets frequently use the word. This process emphasizes words that are significant to the tweet. Table 7 shows terms with the smallest and largest Tf-Idf values. This, using bigram normalized tokenization with stop words included and mindf = 2:

Table 7. Sorted features with smallest and largest Tf-Idf values.

Tf-Idf Values	Features
Smallest	'buen lunes', 'app', 'inicio semana', 'inmediato', 'oms', 'periodico hoy'
	'alto contagio', 'ganar seguidor', 'calidad', 'amlolujo'
Largest	'financiero', 'muerte covid', 'lugar', 'movil', 'movilidad', 'dar positivo'
	'lopez', 'muerte', 'cuidarte profesional', 'gracia'

4.2.2. Word Embedding and Phrase Modeling

Word embedding is a technique for representing words as vectors. The purpose is to reduce the high-dimensional word features to low-dimensional feature vectors while preserving the context similarity.

1. Word2Vec: is a popular natural language processing technique that represents words as dense vectors in a continuous vector space. It relies on the assumption that words with similar meanings often appear in similar contexts. Word2Vec uses a neural network to learn these word embeddings, capturing semantic relationships and similarities between words [59]. This allows for word representations that can be used in various NLP tasks like sentiment analysis, machine translation, and information retrieval. CBOW (Continuous Bag of Words) and Skip-gram are the two fundamental architectures in Word2Vec. CBOW aims to predict a target word based on its surrounding context words. Skip-gram predicts the context words given a target word. Both architectures contribute to creating meaningful word embeddings. In the Gensim library, we can specify whether to use CBOW or Skip-gram. We decided to use the Skip-gram technique since it performed better in our experiments.
2. Doc2Vec: extends the principles of Word2Vec to generate fixed-length vector representations for entire documents, such as sentences, paragraphs, or even entire documents. Doc2Vec employs neural networks to learn these document embeddings while considering the context of words within the document. It assigns a unique vector to each document, capturing its semantic content, and allowing for similarity comparisons between documents [60].
3. Phrase Modeling: The Gensim library offers phrase detection, similar to the *n*-gram representation. However, instead of getting all *n*-grams by sliding the window, it detects frequently used phrases and sticks them together. Hence, we integrated the vector representation of sentences to capture the collective meaning of a group of words, rather than merely aggregating the meanings of individual words. This process allows us to extract many reasonable phrases while keeping the vocabulary size in a manageable size [59]. We built 2-gram and 3-gram models to detect and combine frequently used two and three-word phrases within the corpus. After we obtained the corpus with the phrases, we did the same Doc2Vec process previously applied to unigram tokens. Thus, we present the results for each, using both the Distributed Bag of Words (DBOW) and Distributed Memory (DM), as well as their combination. DBOW and DM are two distinct training algorithms used to generate vector representations of documents.

Table 8 shows the phrases detected by Gensim's phrase detection algorithm in a given tweet. We can see that unigram yields 13 tokens, while 2-gram and 3-gram yields 10 and 9 tokens, respectively. As we can see, phrases like *'no es justo'* and *'quedate en casa'* become a trigram but the rest of the tokens remain as unigrams. This is because the algorithm only extracts the most significant *n*-grams.

Table 8. Phrase detection tokens yield by each model.

	Phrase Detection
Unigram	['@usuario', 'por', 'su', 'trabajo', 'no', 'es', 'justo', 'para', 'los', 'demas', 'quedate', 'en', 'casa']
Bigram	['@usuario', 'por', 'su trabajo', 'no es', 'justo', 'para', 'los', 'demas', 'quedate', 'en casa']
Trigram	['@usuario', 'por', 'su', 'trabajo', 'no es justo', 'para', 'los', 'demas', 'quedate en casa']

4.2.3. Spanish Pre-Trained BERT Models

BERT models are considered state-of-the-art models for various NLP tasks that involve text representation. BERT possesses the significant benefit of supporting transfer learning. These language models have undergone extensive training over several days on robust machines over a large amount of text from platforms like Wikipedia and news websites. A pre-trained model can then be fine-tuned to align with our specific classification task.

We used some variants of BERT models in Spanish. These are particularly useful since they have been built for NLP tasks with high-dimensional analysis. Spanish models are hard to come by, and when available, they are frequently developed using substantial proprietary datasets and resources. As a result, the relevant algorithms and techniques are restricted to large technology corporations. However, a fundamental objective of these models is to promote openness by making them available as open-source resources. Examples of these models are

1. BERTIN: Series of BERT-based models in Spanish, the current model hub points to the best of RoBERTa-base models trained from scratch on the Spanish portion of mC4 using Flax—a neural network library ecosystem for JAX designed for flexibility [61].
2. RoBERTuito: A language model for user-generated content in Spanish, trained following the RoBERTa guidelines on 500 million tweets. RoBERTuito comes cased and uncased [62].
3. BETO: Another model trained on a large Spanish corpus that is size similar to a BERT-Base and was trained with the whole word masking technique, which outperforms some other models [63].

In recent years, there have been more advances in pre-trained BERT models [64]. As a result of their growing popularity, several versions of lighter and faster versions of BERT (e.g., DistilBERT) have been made available to accelerate training and inference processes. However, there is a lack of these for languages other than English.

4.3. Models and Algorithms

4.3.1. Ready-to-Use Libraries

We evaluated NLP libraries that do not need to train machine learning-specific models:

- TextBlob: A Python library that allows users to perform various textual data processing tasks. For sentiment analysis, this tool uses a lexicon-based approach. It makes use of a vocabulary consisting of around 3000 words in English along with their corresponding scores. Thus, for a given text, the TextBlob sentiment analyzer returns two outputs. The polarity value belongs to $[-1, 1]$, where -1 indicates a negative sentiment text, and +1 a positive one. On the other hand, the subjectivity value ranges from 0 to 1, with 0 indicating an objective text, while 1 representing a subjective text. Table 9 shows examples of how TextBlob works with different sentences in Spanish. Prior analysis, these sentences were previously translated into English.
- VADER (Valence Aware Dictionary and sEntiment Reasoner): Similar to TextBlob, this tool uses a sentiment analyzer that is based on a lexicon. But, this tool is specifically tuned to the sentiments expressed in social media since its lexicon (of approximately 9000 token features) includes slangs and emoticons [65]. The words in the lexicon

have a valence score that ranges from extremely positive [4] to extremely negative [−4], with [0] representing neutral sentiment. These scores are determined based on the semantic orientation of the lexical features. The Table 10 illustrates the functioning of VADER. The first column displays the input text. The 'compound' column shows the normalized sum of the valence scores of each word in the text. A value of [−1] indicates a negative sentiment, while [+1] indicates a positive polarity in the text. The columns 'neg', 'neu', and 'pos' indicate the percentage likelihood of the text belonging to the negative, neutral, or positive class, respectively. Vader has achieved good results in English texts due to the quality of its lexicon [66]. However, our texts are in Spanish, so we need a Spanish lexicon (and this resource is not easy to obtain specifically tailored to Latin Spanish and social media) or to translate the tweet into English. We opted for the SentiSense Lexicon [67] which contains a list of Spanish words classified according to their emotional connotation and information about the intensity of the emotion transmitted by each word.

- Pysentimiento Multilingual Toolkit: A very useful transformer-based library for Text Mining and Social NLP tasks such as sentiment analysis and hate speech detection [68] for text classification in Spanish this library uses BETO (https://github.com/dccuchile/beto (accessed on 20 February 2024)) and RoBERTuito (https://github.com/pysentimiento/robertuito (accessed on 20 February 2024)) language models. Pysentimiento is trained with *'pos'*, *'neg'*, *'neu'* labels using the *'TASS-2020 task-1'* corpus (http://tass.sepln.org/2020/ (accessed on 20 February 2024)) merged with the Spanish subsets for each dialect, summing up to 6000 tweets.

Table 9. TextBlob outputs for different statements in Spanish.

Input	'polarity'	'subjectivity'
Este teléfono tiene una pantalla de excelente resolución, además es muy rápido	0.63	0.89
Este teléfono tiene una pantalla de alta resolución, además es rápido	0.18	0.57
Este telefono es lo máximo, lo adoro <3 :D	1.0	1.0
Este telefono no me gusta :(	−0.75	1.0

Table 10. Vader outputs for different statements (in Spanish).

Input	'neg'	'neu'	'pos'	'compound'
hoy es un pésimo día	0.779	0.221	0.0	−0.5461
hoy es un mal día	0.646	0.354	0.0	−0.7424
hoy es un día cualquiera	0.123	0.637	0.24	0.231
hoy es un gran día	0.0	0.408	0.592	0.5404
hoy es un excelente día	0.0	0.294	0.706	0.8633

4.3.2. Machine Learning Algorithms

We also trained and evaluated supervised classification models with the SENT-COVID corpus. This was done under the hypothesis that the models trained with the corpus outperform the ready-to-use libraries. The algorithms we evaluated were:

- Logistic Regression: A generalized linear model widely employed in machine learning applications for classification purposes. It is especially useful for text mining tasks because of its ability to handle large, sparse data sets with robust performance [69]. Logistic regression is used to compress the output of a given set of data into discrete values to a categorical response value. As $\hat{y}$ output is the probability that the input instance belongs to a certain class, we use a binary 'one vs. the-rest' model for each

class. This is interpreted as the probability of being or not being within the class. Hence, three binary classifiers ['neg', 'neu', 'pos'] are created, which we trained with the following parameters using the scikit-learn library:

1. Regularization: We use *'L2'* penalty (this is by using the usual Euclidian distance when calculating the norm) on estimated coefficients (as Ridge regression), which can be controlled using the '*C*' parameter. Higher values of '*C*' correspond to reduced regularization, allowing the model to prioritize fitting the training data optimally. In contrast, for lower values, the model gives priority to finding coefficients close to zero, even if this means a slightly reduced fit to the training data.
2. Multi_class: The training algorithm uses the one-vs-rest scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'.
3. Solver: Algorithm to use in the optimization problem. Only 'newton-cg', 'sag', and 'lbfgs' solvers support L2 regularization with primal formulation as we selected for penalty.
4. Formulation: (Dual is only implemented for L2 penalty with 'liblinear' solver) We prefer Primal when training data instances are greater than the number of features and Dual for other cases.

- Naive Bayes: Naive Bayes (NB) classifier is a probabilistic classifier based on Bayes's theorem for prior distributions. It is applied in problems such as spam filtering, text classification, and hybrid recommender systems [69]. The NB classifier has been shown to be optimal and efficient in many machine learning text classification tasks (especially with independence between document labels assumptions) [70,71]. According to our needs, we decided to use the version of Multinomial Naive Bayes which is used specifically for discrete cases (such as word counts in documents) and incorporates the assumption that characteristics follow a multinomial distribution. In sentiment analysis this model can work better since it allows text data modeled as word frequencies to be handled more efficiently and this makes it particularly useful. The BoW model is used as a feature model for implementation because it has been found to produce results comparable to those obtained by Support Vector Machines and logistic regression algorithms. The predicted label $\hat{y}$ is the y that maximizes the probability of Y given X. We considered the following parameters for NB:

  1. alpha: We conducted tests using the smoothing parameter, which has a default value 1.
  2. force_alpha: If 'False' is set and alpha is less than 10^{-10}, it sets alpha to 10^{-10}. If 'True', alpha remains unchanged. In this case, if alpha is too close to zero, it may cause numerical errors. Besides, when alpha = 0 and force_alpha = true means no smoothing.
  3. fit_prior: Whether to learn class prior probabilities or not. If false, a uniform prior is used.
  4. class_prior: The prior class probabilities of the model. If not specified, these priors are adjusted according to the data frequency. This is useful for an imbalanced class distribution.

- Support Vector Machines (SVM): Various studies show that SVM outperforms other classification algorithms [72] for text classification problems. The SVM objective (primal) is to find the decision surface that maximizes the margin between the data points from different classes. In the linear case, this classifier rewards the amount of separation between classes by applying a sign function to produce a categorical output. Consequently, we handle multi-class classification by creating single linear binary classifiers for each class. Once this criterion has been defined as a decision rule, we define the decision boundaries and corresponding classification margins for each

classifier. The most proficient classifier is the 'linear support vector machine', which is characterized by the maximum margin of separation between points.
The advantage of this method is that slight modifications to the data for a particular document will not alter the label that the classifier assigns. So, the approach is more resistant to noise or perturbations. Also, it is still effective even when the number of features is greater than the number of data instances, and it only requires a limited number of training data to learn the decision function, making it memory efficient.
We chose LinearSVC in scikit-learn rather than SVC implemented in terms of liblinear rather than libsvm. So, the choice of penalties and loss functions has more flexibility and scales better to large numbers of samples. Thus, we tested the following parameters:

1. Regularization: We applied *L2* penalty with *'C'* = 1 same as in logistic regression. In order to determine the significance of correctly labeling individual documents, smaller values of 'C' (more regularization) indicate a greater tolerance for errors on individual documents.
2. Kernel: A linear kernel usually works better when using text data, but we also tested polynomial kernels.
3. Multi_class: We preferred learning fewer number of classifiers so we used one-versus-rest over one-versus-one.

- Multilayer Perceptron (MLP): It is one of the simplest neural network models. These networks have achieved remarkable results on various classification problems, from object classification in images to fast, accurate machine translation [73]. Their approach is similar to logistic regression but takes a step beyond by adding 'hidden layers' contained by 'hidden units'. Each layer performs a non-linear transformation (called activation functions) in the input features. These functions adopt 'S-shaped' curves, and various forms of them were considered during data training. Introducing this extra hidden layer renders the prediction model more complex compared to logistic regression. However, the computational cost is greater, as predicting the response necessitates computing a distinct initial weighted sum of feature values for each hidden unit.

  1. Hidden layer sizes: A list with one element for each hidden layer that gives the number of hidden units for that layer. We passed two values of 100 (two hidden layers, 100 units per layer)
  2. Activation function: Activation function for the hidden layer. Options include the logistic sigmoid, hyperbolic tan, or rectified linear unit functions. In this work, we used the rectified linear unit function.
  3. Solver: The solver for weight optimization. 'lbfgs' is an optimizer in the family of quasi-Newton methods, 'sgd' refers to stochastic gradient descent, 'adam' refers to a stochastic gradient-based optimizer proposed by Kingma, Diederik [74]. The default solver 'adam' works pretty well on relatively large datasets in terms of both training time and validation score.

4.3.3. Transformers

With their self-attention mechanisms, transformer-based models have brought a paradigm shift in natural language processing (NLP) [75]. Unlike traditional recurrent or convolutional neural networks, transformers can process entire sequences of input tokens in parallel. This parallel processing enables efficient computation of contextual representations. The multi-head self-attention mechanisms empower the model to assign importance to each token in the input sequence based on its contextual relevance to other tokens. This feature allows transformers to effectively capture long-range dependencies and contextual information, making them an ideal choice for tasks that demand an understanding of complex linguistic patterns, such as sentiment analysis. To use transformer-based models for sentiment analysis in Spanish, one typically fine-tunes a pre-trained transformer model

on a labeled dataset of Spanish text. The pre-training process involves initializing the model's parameters with weights learned from a large corpus of Spanish text. During fine-tuning, the model learns to adjust its parameters better to capture the nuances of sentiments, leveraging the contextual information encoded in the transformer's self-attention mechanisms. Once the model is fine-tuned, it can be used to predict the sentiment of new text inputs by feeding them through the model and interpreting the output probabilities or scores assigned to each sentiment class.

Using transformer pre-trained models is computationally expensive in training, limiting our ability to execute comprehensive cross-validation tests for score metrics or conduct extensive hyperparameter searches. The experiments were carried out with a single train-test split of the data, usinng 75% for training and 25% for testing. We created the neural network with a single hidden layer and a single output unit. Essential parameters such as input size, hidden units, output size, batch size, dropout, and learning rate were considered. Then, we randomly initialized the dummy input and the output target data (or tensor), and using built-in functions, we created a simple sequential model with an output sigmoid layer and defined the corresponding loss function. This computed the mean-squared error between the input, target, and optimizer. For the gradient descent, the 'torch.optim' package provides various optimization algorithms, so we used a stochastic gradient descent (SGD) optimizer. Finally, we defined the training loop with the following steps:

- Forward propagation: This computed the predicted $\hat{y}$ label and calculated the current loss. This helped us see how the model trains over each epoch (we considered 5 epochs).
- Backward propagation: After each epoch, we set the gradients to zero before starting.
- Gradient descent: Finally, we updated model parameters by calling the optimizer function.

5. Results of Sentiment Analysis Methods

We split the SENT-COVID corpus in a single train-test division randomly separated into approximately 3600 for training and 1200 for testing (75–25%) and used the same seed to preserve the same partitions in different experiments. Table 11 shows the distribution of labels in each partition. It can be observed that both partitions maintain the same class distribution.

Table 11. Distribution of labels in the train and test partitions.

(Seed = 37)	Negative	Neutral	Positive
Train	33.642%	44.934%	21.422%
Test	34.899%	44.380%	20.713%

We established a fundamental baseline for benchmarking performance among the different classification methods. For this, we used the Zero Rule (ZeroR), which predicts the most frequent class in the training dataset. If a model's performance is worse than ZeroR under the same parameters, it suggests the model is useless. As shown in Table 11 the majority class is neutral with 44%. This implies that a ZeroR classifier that predicts the neutral class consistently for each test data point would achieve an accuracy rate of 44%.

The initial experiments were conducted using the BoW model to determine the best vocabulary size for the classification models. Subsequently, a second set of experiments was carried out using word embeddings, Doc2Vec models, and phrase modeling instead of the classical BoW model. The goal was to compare different text representations and their usefulness in machine learning algorithms.

5.1. Vocabulary Size Evaluation

Initially, We wanted to figure out how many features are suitable for the model and seek insights that could guide the establishment of a simplistic criterion for feature selection.

However, since the selection of text features is a broad topic, we do not address it in this paper. An attempt to preserve some of the contexts lost when using BoW model was done by considering the use of n-grams and removing stop words (or just some of them using the frequency of use as a criterion). Thus, we tested different numbers of n-grams and stopwords comparing them using simple logistic regression and computing the accuracy on the test set for the different vocabulary sizes.

In addition to n-grams and stopwords, we compared the results produced by the TF and Tf-Idf weighting schemes. Then, these findings were combined with the previous experiments, aligning with various text processing approaches such as lemmatization, stemming, and normalization applied to tweets.

As we see in Figure 2a the prediction accuracy is improved when more features are included in the model. Interestingly, removing the 'stopwords' is not useful for increasing the accuracy of the model even though these words do not carry semantic information. Figure 2b shows that using 'unigrams' (i.e., bag of words) works better as features increase than using 'bigrams' or 'trigrams'. This means that the n-grams were unable to capture the desired context, so tokens of just one word seem to do the work better for sentiment classification.

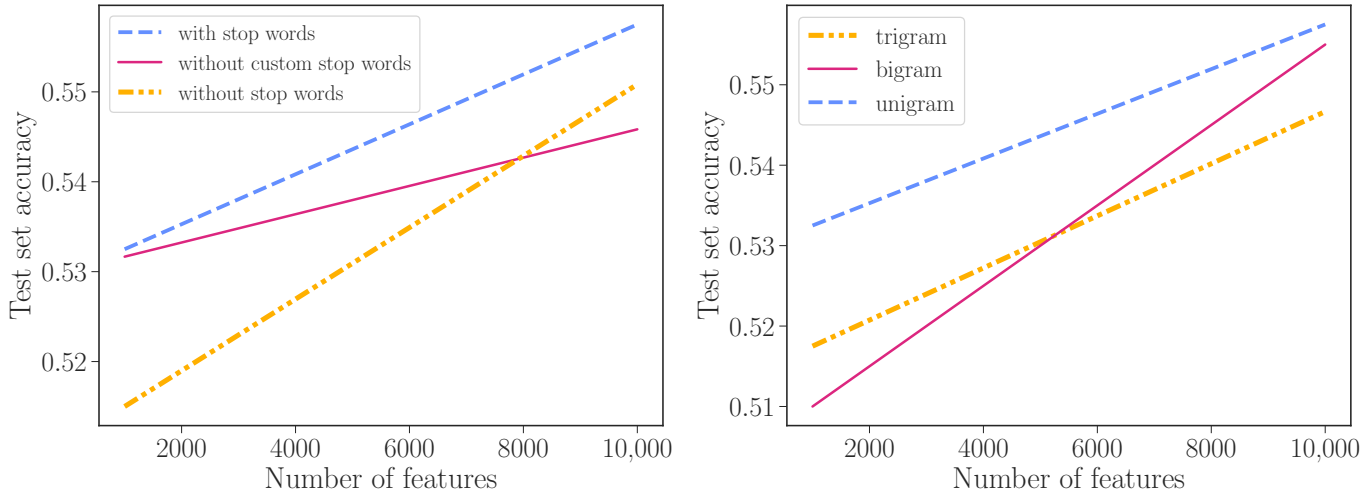


Figure 2. Test Accuracy for different number of features. (**a**) Without vs with stopwords using unigrams. (**b**) n-gram test results. We tested $N = 1, 2, 3$.

5.2. Dimension Reduction Evaluation

When constructing a BoW model, we encounter a high number of features, necessitating a reduction in feature dimensions prior to their integration into learning models. We explored several feature selection methods for comparative analysis in our experiments.

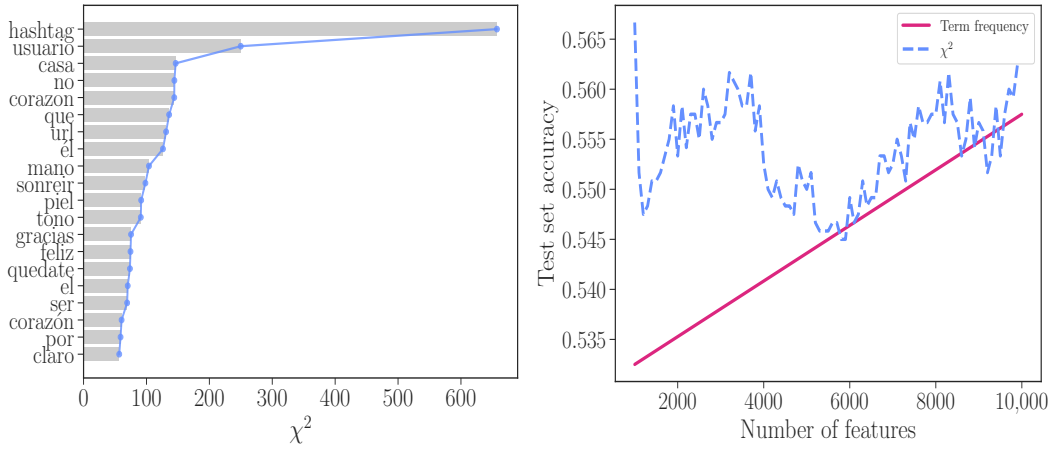
5.2.1. χ^2 Feature Selection

The chi-squared (χ^2) statistic measures the degree of dependence between a feature (here, a term within a tweet) and a class (the sentiment of the tweet, whether positive or negative). Through a contingency table, which shows frequency distribution, we see the relationship between a term within a tweet and the class that the tweet belongs to.

Initially, we assessed this method using the BoW model. This involved transforming the training data into TF vectors, followed by the calculation of the χ^2 statistics between each feature and class. This score helps to select the number of features with the highest values relative to the classes. Subsequently, we used the χ^2 statistic to determine which features were useful and then presented our findings graphically to show which word features were important for prediction. For better visualization, only the top 20 features are

shown in Figure 3. We then decreased the dimensions to different amounts of features and assessed the precision based on the test set.

Figure 3a shows the 20 most significant words identified, some of which are surprisingly considered 'stopwords'. The plot in Figure 3b shows that enhanced accuracy was achieved by selecting around 8000 features based on the χ^2 criterion rather than the unconstrained BoW model. Despite the slight increase in accuracy, the main objective of dimension reduction has not been completely achieved. Fewer features do not necessarily obtain better results. But we can see that using 3000 χ^2 selected features (0.56 of accuracy) yields better results than employing the most frequent 9000 features (0.55 of accuracy).



(**a**) (**b**)

Figure 3. (**a**) Most significant words given by χ^2 and (**b**) accuracy on the test set for the different number of features. We show results for the term frequency vector reduced by the term frequency (solid line) and the χ^2 (dashed line).

5.2.2. Truncated Singular Value Decomposition

Another method to reduce the dimension is Singular Value Decomposition (SVD). Contrary to the Principal Component Analysis (PCA) method, this estimator does not center the data before computing the singular value decomposition. In SVD the term-document matrix is decomposed into three matrices—U, σ (sigma), and V—and retaining the top-k singular values and their corresponding columns from U and V [76]. These retained singular vectors effectively represent the most important features in the text data. Subsequently, these vectors can be utilized as a reduced feature set of n components.

In Figure 4, we observe that with a minimum of 1000 components, more than 90% of the variance is already taken into account, which is a considerable reduction. To determine whether these new features yield good predictions, we tested the accuracy as we have done in previous experiments. Table 12 shows that the best results are obtained when using around 2000 components.

Table 12. Accuracy for n components.

n_components	Accuracy
1000	63.12%
1500	64.79%
2000	65.76%
2500	65.51%
3000	64.83%

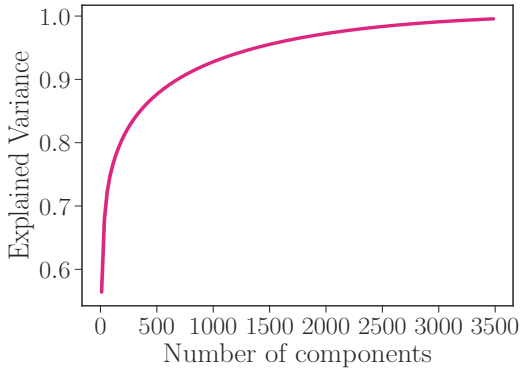


Figure 4. Explained variance for n components.

5.3. Document Embeddings Evaluation

To test this vector representation model, we obtained the 'embeddings' using pre-trained vectors, specifically utilizing the Word2Vec embeddings from Spanish Billion Word Corpora (https://crscardellino.github.io/SBWCE/ (accessed on 20 February 2024)). Employing these 300-size vectors we were able to represent a tweet as a vector in a more precise way through the Doc2Vec models.

Next, regarding the identification of phrases, we initially constructed the '1-gram', '2-gram', and '3-gram' representations of the tokens from all documents. With the Gensim library, we implemented Doc2Vec to learn the paragraph and document embeddings via distributed memory (DM) and distributed Bag of Words (DBOW). Specifically, for DM we used the Distributed Memory Concatenation (DMC) and Distributed Memory Mean (DMM) alternative training algorithms for document vector generation. DMC enhances the DM model by concatenating the document vector with the average of context word vectors, aiming to capture both overall document semantics and specific word context. On the other hand, DMM simplifies this process by directly averaging the context word vectors without concatenation. Once this was done, we compared them evaluating the accuracy on the test set. These methods were tested both separately and in combination.

In Table 13 we can see that the best results are given when combining the DBOW and DMM models. Although these results are not better than what we have obtained so far, it is remarkable that the representation using '2-gram' and '3-gram' was increasingly effective. This can potentially be a direction to explore further.

Table 13. Test accuracy for Doc2Vec models. The best result is highlighted in bold.

	1-Gram	**2-Gram**	**3-Gram**	**Best**
DBOW	60.642%	59.934%	60.422%	60.642%
DMC	56.893%	54.387%	55.713%	56.893%
DMM	59.641%	58.935%	57.253%	59.641%
DBOW+DMC	61.927%	61.234%	62.422%	62.422%
DBOW+DMM	63.185%	62.617%	63.373%	**63.373**%

5.4. Hyperparameter Tuning

Having obtained representations of the corpus suitable for learning models, we shifted our focus to model selection and hyperparameter optimization to get the best performance and accuracy in our prediction. Our decisions were driven by optimizing accuracy. Therefore, we performed a 10-fold grid search cross-validation method using a repeated stratified k-fold to identify the optimal values within the range we examined for each classification algorithm. For logistic regression, we explored a range of regularization strength (C) that spans from 10^{-5} to 10^{2} (or from 0.00001 to 100) on a logarithmic scale. For the solvers, we consider {'newton-cg', 'lbfgs', 'liblinear'}, and for specifying the penalty we experimented

with the values {'none', 'l1', 'l2', 'elasticnet'}. The Naïve Bayes model was tested with the smoothing parameter alpha varying twenty values from 0 to 1, with both true and false fit_prior. For the implementation of SVM, we used the same set of regularization strength values as those used in logistic regression. Furthermore, we conducted trials using {'poly', 'rbf', 'sigmoid' } kernels with their default coefficients. Lastly, when creating our neural network architecture, we explored 1 to 5 hidden layer sizes. We also tested different activation functions including {'relu', 'tanh', 'logistic'}, and considered multiple solvers like {'lbfgs', 'sgd', 'adam'}. Table 14 summarizes the values tested for the different models.

In Table 14, the chosen parameters for each supervised learning algorithm are presented as the optimal ones, utilizing the BoW unconstrained model as the feature set. Regrettably, there is little variation in performance when employing various penalty values or types, showing only slight enhancements. This observation also holds for the alpha smoothing parameter within the Naive Bayes model.

Table 14. Optimal hyperparameters settings selected for each model based on th optimization of accuracy through grid-search and cross-validation.

GridSearchCV (CV = 10)		
Model	**Hyperparameters Tested**	**Optimal Value**
Logistic Regression	C :[10^{-5}, 10^{-4}, 10^{-3}, 0.01, 0.1, 1, 10, 10^2], [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] solver :['newton-cg', 'lbfgs', 'liblinear'] penalty :['none', 'l1', 'l2', 'elasticnet']	C : 0.7, solver : 'lbfgs', penalty : 'l2'
Multinomial Naive Bayes	alpha : 20 values from 0 to 1 fit_prior : ['true', 'false'] class_prior : [None, [0.35, 0.4, 0.25]]	alpha : 0.85, force_alpha : 'true', class_prior : None
SVM	C : [10^{-5}, 10^{-4}, 10^{-3}, 0.01, 0.1, 1, 10, 10^2] kernel : ['linear', 'poly', 'rbf', 'sigmoid'] gamma : ['scale']	C : 1, gamma : 'scale', kernel : 'poly'
Multilayer Perceptron	Layers : from 1 to 5 solver : ['lbfgs', 'sgd', 'adam'] Activation : ['relu', 'tanh', 'logistic']	Layers : 2, solver : 'adam', Activation : 'relu'

5.5. Sentiment Analysis Final Models

Finally, we fitted the supervised learning algorithms using the hyperparameters found with the grid search using as features the unconstrained and reduced BoW representations, as well as the Doc2Vec with DBOW+DMM using '3-gram'. Table 15 reports the results of the obtained classification models in terms of the accuracy, precision, recall and the micro F1-score using 10-fold cross-validation. Additionally, the results of fitting an additional KNN model with k = 5 (which does not estimate coefficients) were only included as a reference point for the performance of the models we analyzed. This non-parametric model was included for empirical analysis purposes. Although we cannot affirm that the parametric models are more effective, initial observations appear to indicate that they are more successful in this task.

For the black-box libraries, there is no need for matrix transformations or data splitting, the evaluation is done over the complete set because there is no need to train an algorithm. We only need to pre-process the corpus. This is especially important for libraries such as Vader, which needs to match the largest number of words possible to obtain better predictions. The same applies to pysentimiento since it is not recommended to use lemmatized words (in fact, low processing is more recommended for this library). Vader has a low computational cost, while pysentimiento requires a bit of time to compute the results,

but the advantage is that no prior knowledge is needed. Table 16 shows the results of sentiment analysis using the black-box libraries. We can see substantial difference in all performance metrics among TextBlob, VADER and Pysentimiento. Given that TextBlob relies on a pattern-based and lexicon-driven approach and VADER, which is tailored for social media texts, utilizes a rule-based system with a sentiment lexicon to capture both polarity and intensity; both fail to capture the nuances of the Spanish language. PySentimiento is specifically designed for Spanish texts. It employs machine learning models trained on large corpora to classify sentiment and detect emotions, offering a more nuanced and context-aware approach compared to the rule-based methods of TextBlob and VADER, particularly in the Spanish language domain.

Table 15. Results of the obtained by training the classification algorithms on the three feature sets. The best result is highlighted in bold.

	Unconstrained BoW				
	KNN	**SVM**	**MNB**	**Logistic**	**MLP**
Accuracy	59.75%	62.31%	62.03%	64.26%	63.51%
Precision	60.08%	62.25%	63.24%	64.39%	63.92%
Recall	59.76%	62.31%	62.03%	64.25%	63.51%
F1-score	58.78%	61.96%	60.78%	63.61%	62.59%
	BoW reduced with SVD				
	KNN	SVM	MNB	Logistic	MLP
Accuracy	60.88%	64.98%	64.19%	**67.12**%	68.84%
Precision	61.40%	64.91%	62.27%	66.91%	67.25%
Recall	60.88%	64.98%	64.19%	67.12%	68.84%
F1-score	59.92%	64.02%	62.98%	66.24%	67.90%
	Doc2Vec with DBOW+DMM				
	KNN	SVM	MNB	Logistic	MLP
Accuracy	56.47%	62.56%	62.96%	63.68%	64.31%
Precision	54.76%	63.81%	63.54%	61.94%	60.81%
Recall	56.47%	62.56%	62.96%	63.68%	62.31%
F1-score	42.93%	57.59%	63.88%	61.55%	62.11%

Table 16. Results obtained by the sentiment analysis libraries. The best result is highlighted in bold.

	TextBlob	**Nltk Vader**	**Pysentimiento**
accuracy	51.23%	58.07%	**68.89**%
precision	55.45%	58.60%	72.20%
recall	51.23%	57.19%	52.81%
F1-score	52.92%	56.42%	60.38%

Table 17 shows the evaluation of the Spanish BERT models, BETO, BerTin-base, and roBERTa. All of them had been pre-trained in the Spanish language, with the latter specifically for social networks (robertuito). We employed their pre-trained weights to tokenize the text using the corresponding tokenizers for each model. Additionally, we used the PyTorch library with a batch data size of 16 and tested with the unconstrained BoW model since pre-trained models have shown to perform better with all the words within the tweet present. We have defined a loss function based on Categorical Cross-Entropy which measures the discrepancy between the probability distribution predicted and the real probability distribution of the labels, in this way we can calculate the loss in each iteration of the loop, then the calculated loss is used to perform gradient backpropagation to adjust the model parameters during training. A dropout was also defined to regularize the model and prevent overfitting and prevent neurons from becoming too dependent on each other during training, in this way, a dropout layer is created that specifies the probability that a

neuron is deactivated during training. the training. In this case, it is set to 0.3, which means that each neuron has a 30% probability of being deactivated during each training step.

Finetunning on the pre-trained models yielded the best results. However, the more epochs we use for training, the more overfitting we observe, as it can be seen in Table 18, where we show the accuracy on the train and test sets when training on 3, 5 and 10 epochs. This increase in variance is not as easy to interpret as the training error from previous models, where we can take a better look at error rates.

Table 17. Classification results of the Spanish BERT models. The best result is highlighted in bold.

	BETO-Uncased	**roBERTa-Sentiment**	**BerTin-Base**
training set accuracy	96.20%	97.54%	96.91%
validation set accuracy	**73.26**%	71.88%	72.14%
validation loss	0.3945	0.2847	0.2141

Table 18. Results of an increasing number of epochs using BETO. The best result is highlighted in bold.

Epoch	**Train Set Accuracy**	**Test Set Accuracy**	**Validation Loss**
3	92.89%	70.33%	0.4554
5	96.20%	**73.26**%	0.3945
10	97.12%	72.76%	0.3161

Finally, Table 19 compares the best accuracy scores of various sentiment models evaluated in our study. In particular, the table shows that BERT-based models perform better than ruled-based or machine learning models. These levels of accuracy are even achieved by the ready-to-use Pysentimiento library. Among these, BETO-uncased is the most accurate model, with an accuracy score of 73.26%. This performance of the BERT-base models may be due to the fact that the model achieves a deep understanding of the context and nuances of the Mexican Spanish discourse, acquired through an extensive pre-training on a variety of text corpora. This allows them to capture the expressions of sentiment and complex linguistic structures of COVID-19 related tweets. In contrast, rule-based models such as Vader and TextBlob exhibit the lowest accuracy scores in our analysis, reflecting their limitation to adapting to the specific linguistic structure and domain.

Table 19. Summary of the performance evaluated based on the accuracy of different sentiment analysis models on the SENT-COVID corpus. The best result is highlighted in bold.

Model	**Accuracy**
TextBlob	51.23%
Nltk Vader	58.07%
Pysentimiento	68.89%
SVM	64.89%
Naive Bayes	62.22%
Logistic Regression	67.12%
MLP	68.84%
BETO-uncased	**73.26**%
roBERTa-sentiment	71.88%
BerTin-base	72.14 %

6. Conclusions

This paper presents SENT-COVID, a Twitter corpus of COVID-19 in Mexican Spanish manually annotated with polarity. We have designed several classification experiments with this resource using ready-to-use libraries, classical machine learning methods, and deep learning approaches based on transformers.

In light of the temporal context surrounding the compilation and presentation of our corpus, it is crucial to emphasize the importance of its value in hindsight. While we acknowledge that the corpus's arrival may seem overdue, We firmly assert that it remains relevant to our understanding of linguistic patterns and public discourse. As a historical archive of Mexican Spanish tweets during the pandemic, our corpus offers unique insights into the evolution of societal responses, linguistic shifts, and sentiment fluctuations over time. Despite the availability of other resources, the retrospective nature of our corpus provides researchers with an invaluable opportunity to conduct comparative analyses, trace the trajectory of linguistic trends, and evaluate the enduring impact of COVID-19 discourse on societal norms and behaviors. Furthermore, we emphasize the corpus's potential to complement existing datasets and tools, enriching interdisciplinary research endeavors in fields such as linguistics, public health communication, and computational social science.

Given the experiments, we observe that, among the black-box libraries, neither TextBlob nor Vader demonstrated satisfactory performance, probably due to the difficulty of obtaining a suitable lexicon in Spanish. In contrast, Pysentimiento exhibits better performance because it employs machine learning models trained on large Spanish corpora to classify text into sentiment categories such as positive, negative, or neutral, and to detect emotions such as joy, anger, sadness, and fear with higher accuracy and contextual understanding. By leveraging machine learning techniques, PySentimiento can capture the nuances of sentiment expressed in Spanish text more effectively, overcoming the limitations faced by lexicon-based approaches like TextBlob and Vader.

The supervised models have revealed that contrary to our initial expectations, removing common words is not as effective as we had thought. However, the models showed that including a broader range of features and observations improved performance without requiring too much computing power. The dimension reduction models managed to improve the prediction results with fewer features, so we can conclude that it is a viable alternative to tackle this problem. However, there is still much to explore. Furthermore, the penalty parameter selection did not make a major difference as expected, neither Ridge nor Lasso regularization, and it performed almost the same as with the default parameters.

The results of the Doc2Vec models did not meet the expectations, as they could not outperform basic BoW models. Additionally, training these models is associated with a higher computational cost.

Finally, pre-trained BERT models yielded the best results. However, they are the most expensive in terms of computational cost. Additionally, it is difficult to perform different tests since cross-validation is difficult. Therefore, the parameters and configuration settings must be chosen based on another criterion. Despite these challenges, for datasets that are not too large, pre-trained BERT models are the most suitable choice.

Author Contributions: Conceptualization, H.G.-A., G.B.-E. and G.S.; methodology, H.G.-A., G.B.-E. and G.S.; software, H.G.-A. and J.-C.B.; validation, G.B.-E., G.S. and W.Á.; formal analysis, H.G.-A. and J.-C.B.; investigation, H.G.-A. and J.-C.B.; resources, H.G.-A., G.B.-E. and G.S.; data curation, H.G.-A., J.-C.B. and W.Á.; writing—original draft preparation, H.G.-A. and J.-C.B.; writing—review and editing, G.B.-E., G.S., and W.Á.; visualization, J.-C.B. and W.Á.; supervision, H.G.-A.; project administration, H.G.-A.; funding acquisition, H.G.-A., G.B.-E. and G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by CONAHCYT project number CF-2023-G-64, and by PAPIIT projects TA101722 and IN104424.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: SENT-COVID corpus can be found here: https://github.com/GIL-UNAM/SENT-COVID (accessed on 20 February 2024). The dataset is licensed under CC0, so it is open data. If the data are used, we would appreciate citing this article as the corpus descriptor.

Acknowledgments: Authors thank CONAHCYT for the computing resources provided through the Deep Learning Platform for Language Technologies of the INAOE Supercomputing Laboratory, as well as Gabriel Castillo for the computing services.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NLP	Natural Language Processing
ML	Machine Learning
DL	Deep Learning
LSTM	Long Short Term Memory
BERT	Bidirectional Encoder Representations from Transformers
IAA	Inter-Annotator Agreement
BoW	Bag of Words
Tf-Idf	Term Frequency-Inverse Document Frequency
MLP	Multilayer Perceptron
CBOW	Continuous Bag of Words
DBOW	Distributed Bag of Words
DM	Distributed Memory
NB	Naïve Bayes
SVM	Support Vector Machines

References

1. Shivaprasad, T.; Shetty, J. Sentiment analysis of product reviews: A review. In Proceedings of the 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 10–11 March 2017; pp. 298–301.
2. Das, A.; Gunturi, K.S.; Chandrasekhar, A.; Padhi, A.; Liu, Q. Automated pipeline for sentiment analysis of political tweets. In Proceedings of the 2021 International Conference on Data Mining Workshops (ICDMW), Auckland, New Zealand, 7–10 December 2021; pp. 128–135.
3. Man, X.; Luo, T.; Lin, J. Financial sentiment analysis (fsa): A survey. In Proceedings of the 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), Taipei, Taiwan, 6–9 May 2019; pp. 617–622.
4. Shelar, A.; Huang, C.Y. Sentiment Analysis of Twitter Data. In Proceedings of the 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 12–14 December 2018; pp. 1301–1302. [CrossRef]
5. Zahoor, S.; Rohilla, R. Twitter Sentiment Analysis Using Lexical or Rule Based Approach: A Case Study. In Proceedings of the 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 4–5 June 2020; pp. 537–542. [CrossRef]
6. Nair, A.J.; G, V.; Vinayak, A. Comparative study of Twitter Sentiment On COVID-19 Tweets. In Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 8–10 April 2021; pp. 1773–1778. [CrossRef]
7. Diyasa, I.G.S.M.; Mandenni, N.M.I.M.; Fachrurrozi, M.I.; Pradika, S.I.; Manab, K.R.N.; Sasmita, N.R. Twitter Sentiment Analysis as an Evaluation and Service Base On Python Textblob. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1125*, 012034. [CrossRef]
8. Aljedaani, W.; Rustam, F.; Mkaouer, M.W.; Ghallab, A.; Rupapara, V.; Washington, P.B.; Lee, E.; Ashraf, I. Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry. *Knowl.-Based Syst.* **2022**, *255*, 109780. [CrossRef]
9. Pradhan, R. Extracting Sentiments from YouTube Comments. In Proceedings of the 2021 Sixth International Conference on Image Information Processing (ICIIP), Shimla, India, 26–28 November 2021; Volume 6, pp. 1–4. [CrossRef]
10. Sahu, S.; Kumar, R.; MohdShafi, P.; Shafi, J.; Kim, S.; Ijaz, M.F. A Hybrid Recommendation System of Upcoming Movies Using Sentiment Analysis of YouTube Trailer Reviews. *Mathematics* **2022**, *10*, 1568. [CrossRef]
11. Alawadh, H.M.; Alabrah, A.; Meraj, T.; Rauf, H.T. English Language Learning via YouTube: An NLP-Based Analysis of Users' Comments. *Computers* **2023**, *12*, 24. [CrossRef]
12. Anastasiou, P.; Tzafilkou, K.; Karapiperis, D.; Tjortjis, C. YouTube Sentiment Analysis on Healthcare Product Campaigns: Combining Lexicons and Machine Learning Models. In Proceedings of the 2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA), Volos, Greece, 10–12 July 2023; pp. 1–8. [CrossRef]

13. Gupta, S.; Kirthica, S. Sentiment Analysis of Youtube Comment Section in Indian News Channels. In Proceedings of the ICT for Intelligent Systems, Ahmedabad, India, 27–28 April 2023; Springer Nature: Singapore, 2023; pp. 191–200.
14. Melton, C.A.; Olusanya, O.A.; Ammar, N.; Shaban-Nejad, A. Public sentiment analysis and topic modeling regarding COVID-19 vaccines on the Reddit social media platform: A call to action for strengthening vaccine confidence. *J. Infect. Public Health* **2021**, *14*, 1505–1512. [CrossRef]
15. Botzer, N.; Gu, S.; Weninger, T. Analysis of Moral Judgment on Reddit. *IEEE Trans. Comput. Soc. Syst.* **2023**, *10*, 947–957. [CrossRef]
16. Ruan, T.; Lv, Q. Public perception of electric vehicles on Reddit and Twitter: A cross-platform analysis. *Transp. Res. Interdiscip. Perspect.* **2023**, *21*, 100872. [CrossRef]
17. Sekar, V.R.; Kannan, T.K.R.; N, S.; Vijay, P. Hybrid Perception Analysis of World Leaders in Reddit using Sentiment Analysis. In Proceedings of the 2023 International Conference on Advances in Intelligent Computing and Applications (AICAPS), Kochi, India, 1–3 February 2023; pp. 1–5. [CrossRef]
18. Ligthart, A.; Catal, C.; Tekinerdogan, B. Systematic reviews in sentiment analysis: A tertiary study. *Artif. Intell. Rev.* **2021**, *54*, 4997–5053. [CrossRef]
19. Shayaa, S.; Jaafar, N.I.; Bahri, S.; Sulaiman, A.; Seuk Wai, P.; Wai Chung, Y.; Piprani, A.Z.; Al-Garadi, M.A. Sentiment Analysis of Big Data: Methods, Applications, and Open Challenges. *IEEE Access* **2018**, *6*, 37807–37827. [CrossRef]
20. Nia, Z.M.; Bragazzi, N.L.; Ahamadi, A.; Asgary, A.; Mellado, B.; Orbinski, J.; Seyyed-Kalantari, L.; Woldegerima, W.A.; Wu, J.; Kong, J.D. Off-label drug use during the COVID-19 pandemic in Africa: topic modelling and sentiment analysis of ivermectin in South Africa and Nigeria as a case study. *J. R. Soc. Interface* **2023**, *20*, 20230200. [CrossRef]
21. Movahedi Nia, Z.; Bragazzi, N.; Asgary, A.; Orbinski, J.; Wu, J.; Kong, J. Mpox Panic, Infodemic, and Stigmatization of the Two-Spirit, Lesbian, Gay, Bisexual, Transgender, Queer or Questioning, Intersex, Asexual Community: Geospatial Analysis, Topic Modeling, and Sentiment Analysis of a Large, Multilingual Social Media Database. *J. Med. Internet Res.* **2023**, *25*, e45108. [CrossRef] [PubMed]
22. Kappaun, A.; Oliveira, J. Análise sobre Viés de Gênero no Youtube: Um Estudo sobre as Eleições Presidenciais de 2018 e 2022. In Proceedings of the Anais do XII Brazilian Workshop on Social Network Analysis and Mining, João Pessoa, PB, Brazil, 6–11 August 2023; SBC: Porto Alegre, RS, Brazil, 2023; pp. 127–138.
23. Aleksandric, A.; Anderson, H.I.; Melcher, S.; Nilizadeh, S.; Wilson, G.M. Spanish Facebook Posts as an Indicator of COVID-19 Vaccine Hesitancy in Texas. *Vaccines* **2022**, *10*, 1713. [CrossRef]
24. Balbontín, C.; Contreras, S.; Browne, R. Using Sentiment Analysis in Understanding the Information and Political Pluralism under the Chilean New Constitution Discussion. *Soc. Sci.* **2023**, *12*, 140. [CrossRef]
25. Agustiningsih, K.K.; Utami, E.; Al Fatta, H. Sentiment Analysis of COVID-19 Vaccine on Twitter Social Media: Systematic Literature Review. In Proceedings of the 2021 IEEE 5th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Purwokerto, Indonesia, 24–25 November 2021; pp. 121–126. [CrossRef]
26. Alamoodi, A.; Zaidan, B.; Zaidan, A.; Albahri, O.; Mohammed, K.; Malik, R.; Almahdi, E.; Chyad, M.; Tareq, Z.; Albahri, A.; et al. Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review. *Expert Syst. Appl.* **2021**, *167*, 114155. [CrossRef]
27. Hussain, A.; Tahir, A.; Hussain, Z.; Sheikh, Z.; Gogate, M.; Dashtipour, K.; Ali, A.; Sheikh, A. Artificial Intelligence–Enabled Analysis of Public Attitudes on Facebook and Twitter Toward COVID-19 Vaccines in the United Kingdom and the United States: Observational Study. *J. Med. Internet Res.* **2021**, *23*, e26627. [CrossRef] [PubMed]
28. Khan, R.; Rustam, F.; Kanwal, K.; Mehmood, A.; Choi, G.S. US Based COVID-19 Tweets Sentiment Analysis Using TextBlob and Supervised Machine Learning Algorithms. In Proceedings of the 2021 International Conference on Artificial Intelligence (ICAI), Islamabad, Pakistan, 5–7 April 2021; pp. 1–8. [CrossRef]
29. Mudassir, M.A.; Mor, Y.; Munot, R.; Shankarmani, R. Sentiment Analysis of COVID-19 Vaccine Perception Using NLP. In Proceedings of the 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2–4 September 2021; pp. 516–521. [CrossRef]
30. Rahul, K.; Jindal, B.R.; Singh, K.; Meel, P. Analysing Public Sentiments Regarding COVID-19 Vaccine on Twitter. In Proceedings of the 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 19–20 March 2021; Volume 1, pp. 488–493. [CrossRef]
31. Abiola, O.; Abayomi-Alli, A.; Tale, O.A.; Misra, S.; Abayomi-Alli, O. Sentiment analysis of COVID-19 tweets from selected hashtags in Nigeria using VADER and Text Blob analyser. *J. Electr. Syst. Inf. Technol.* **2023**, *10*, 5. [CrossRef]
32. Jelodar, H.; Wang, Y.; Orji, R.; Huang, H. Deep Sentiment Classification and Topic Discovery on Novel Coronavirus or COVID-19 Online Discussions: NLP Using LSTM Recurrent Neural Network Approach. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2733–2742. [CrossRef]
33. Chunduri, R.K.; Perera, D.G. Neuromorphic Sentiment Analysis Using Spiking Neural Networks. *Sensors* **2023**, *23*, 7701. [CrossRef]
34. Satu, M.S.; Khan, M.I.; Mahmud, M.; Uddin, S.; Summers, M.A.; Quinn, J.M.; Moni, M.A. TClustVID: A novel machine learning classification model to investigate topics and sentiment in COVID-19 tweets. *Knowl.-Based Syst.* **2021**, *226*, 107126. [CrossRef]
35. Contreras Hernández, S.; Tzili Cruz, M.P.; Espínola Sánchez, J.M.; Pérez Tzili, A. Deep Learning Model for COVID-19 Sentiment Analysis on Twitter. *New Gener. Comput.* **2023**, *41*, 189–212. [CrossRef]

36. Naseem, U.; Razzak, I.; Khushi, M.; Eklund, P.W.; Kim, J. COVIDSenti: A Large-Scale Benchmark Twitter Data Set for COVID-19 Sentiment Analysis. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 1003–1015. [CrossRef] [PubMed]
37. Dimitrov, D.; Baran, E.; Fafalios, P.; Yu, R.; Zhu, X.; Zloch, M.; Dietze, S. TweetsCOV19—A Knowledge Base of Semantically Annotated Tweets about the COVID-19 Pandemic. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 October 2020; Association for Computing Machinery: New York, NY, USA; pp. 2991–2998.
38. Kabir, M.Y.; Madria, S. EMOCOV: Machine learning for emotion detection, analysis and visualization using COVID-19 tweets. *Online Soc. Netw. Media* **2021**, *23*, 100135. [CrossRef] [PubMed]
39. Lamsal, R. Design and analysis of a large-scale COVID-19 tweets dataset. *Appl. Intell.* **2021**, *51*, 2790–2804. [CrossRef] [PubMed]
40. Guo, R.; Xu, K. A Large-Scale Analysis of COVID-19 Twitter Dataset in a New Phase of the Pandemic. In Proceedings of the 2022 IEEE 12th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 15–17 July 2022; pp. 276–281. [CrossRef]
41. Hong, L.; Convertino, G.; Chi, E. Language Matters In Twitter: A Large Scale Study. In Proceedings of the International AAAI Conference on Web and Social Media, Virtually, 7–10 June 2021; Volume 5, pp. 518–521.
42. Lopez, C.E.; Gallemore, C. An augmented multilingual Twitter dataset for studying the COVID-19 infodemic. *Soc. Netw. Anal. Min.* **2021**, *11*, 102. [CrossRef] [PubMed]
43. Imran, M.; Qazi, U.; Ofli, F. TBCOV: Two Billion Multilingual COVID-19 Tweets with Sentiment, Entity, Geo, and Gender Labels. *Data* **2022**, *7*, 8. [CrossRef]
44. Garcia, K.; Berton, L. Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA. *Appl. Soft Comput.* **2021**, *101*, 107057. [CrossRef] [PubMed]
45. Jonker, R.A.A.; Poudel, R.; Fajarda, O.; Matos, S.; Oliveira, J.L.; Lopes, R.P. Portuguese Twitter Dataset on COVID-19. In Proceedings of the 2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Istanbul, Turkey, 10–13 November 2022; pp. 332–338. [CrossRef]
46. Yang, Q.; Alamro, H.; Albaradei, S.; Salhi, A.; Lv, X.; Ma, C.; Alshehri, M.; Jaber, I.; Tifratene, F.; Wang, W.; et al. SenWave: Monitoring the Global Sentiments under the COVID-19 Pandemic. *arXiv* **2020**, arXiv:2006.10842.
47. Al-Laith, A.; Alenezi, M. Monitoring People's Emotions and Symptoms from Arabic Tweets during the COVID-19 Pandemic. *Information* **2021**, *12*, 86. [CrossRef]
48. Balech, S.; Benavent, C.; Calciu, M. The First French COVID19 Lockdown Twitter Dataset. *arXiv* **2020**, arXiv:2005.05075.
49. Babić, K.; Petrović, M.; Beliga, S.; Martinčić-Ipšić, S.; Matešić, M.; Meštrović, A. Characterisation of COVID-19-Related Tweets in the Croatian Language: Framework Based on the Cro-CoV-cseBERT Model. *Appl. Sci.* **2021**, *11*, 442. [CrossRef]
50. Nurdeni, D.A.; Budi, I.; Santoso, A.B. Sentiment Analysis on Covid19 Vaccines in Indonesia: From The Perspective of Sinovac and Pfizer. In Proceedings of the 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), Surabaya, Indonesia, 9–11 April 2021; pp. 122–127. [CrossRef]
51. Samaras, L.; García-Barriocanal, E.; Sicilia, M.A. Sentiment analysis of COVID-19 cases in Greece using Twitter data. *Expert Syst. Appl.* **2023**, *230*, 120577. [CrossRef]
52. Cotik, V.; Debandi, N.; Luque, F.M.; Miguel, P.; Moro, A.; Pérez, J.M.; Serrati, P.; Zajac, J.; Zayat, D. A Study of Hate Speech in Social Media during the COVID-19 Outbreak. 2020. Available online: https://openreview.net/forum?id=01eOESDhbSW (accessed on 15 April 2024).
53. Aragón, M.E.; Jarquín-Vásquez, H.J.; Montes-y Gómez, M.; Escalante, H.J.; Pineda, L.V.; Gómez-Adorno, H.; Posadas-Durán, J.P.; Bel-Enguix, G. Overview of MEX-A3T at IberLEF 2020: Fake News and Aggressiveness Analysis in Mexican Spanish. In Proceedings of the IberLEF@ SEPLN, Virtually, 22 September 2020; pp. 222–235.
54. Vásquez, J.; Andersen, S.; Bel-Enguix, G.; Gómez-Adorno, H.; Ojeda-Trueba, S.L. Homo-mex: A mexican spanish annotated corpus for lgbt+ phobia detection on twitter. In Proceedings of the 7th Workshop on Online Abuse and Harms (WOAH), Toronto, ON, Canada, 13 July 2023; pp. 202–214.
55. Martínez, R.Y.; Blanco, G.; Lourenço, A. Spanish Corpora of tweets about COVID-19 vaccination for automatic stance detection. *Inf. Process. Manag.* **2023**, *60*, 103294. [CrossRef]
56. Plutchik, R. *The Emotions*; University Press of America: Lanham, MD, USA, 1991.
57. Bender, E.M.; Friedman, B. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. *Trans. Assoc. Comput. Linguist.* **2018**, *6*, 587–604. [CrossRef]
58. McHugh, M.L. Interrater reliability: the kappa statistic. *Biochem. Medica* **2012**, *22*, 276–282. [CrossRef]
59. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
60. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning (PMLR), Beijing, China, 22–24 June 2014; pp. 1188–1196.
61. la Rosa y Eduardo G. Ponferrada y Manu Romero y Paulo Villegas y Pablo González de Prado Salas y María Grandury, J.D. BERTIN: Efficient Pre-Training of a Spanish Language Model using Perplexity Sampling. *Proces. Leng. Nat.* **2022**, *68*, 13–23.
62. Pérez, J.M.; Furman, D.A.; Alemany, L.A.; Luque, F. RoBERTuito: A pre-trained language model for social media text in Spanish. *arXiv* **2021**, arXiv:2111.09453.

63. Cañete, J.; Chaperon, G.; Fuentes, R.; Ho, J.H.; Kang, H.; Pérez, J. BETO, Spanish Pre-Trained BERT Model and Evaluation Data. In Proceedings of the PML4DC at ICLR 2020, Virtually, 26 April 2020.
64. Tenney, I.; Das, D.; Pavlick, E. BERT rediscovers the classical NLP pipeline. *arXiv* **2019**, arXiv:1905.05950.
65. 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, Ann Arbor, MI, USA, 1–4 June 2014; Volume 8; pp. 216–225.
66. Pano, T.; Kashef, R. A Complete VADER-Based Sentiment Analysis of Bitcoin (BTC) Tweets during the Era of COVID-19. *Big Data Cogn. Comput.* **2020**, *4*, 33. [CrossRef]
67. de Albornoz, J.C.; Plaza, L.; Gervás, P. SentiSense: An easily scalable concept-based affective lexicon for sentiment analysis. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 21–27 May 2012; Calzolari, N., Choukri, K., Declerck, T., Doğan, M.U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S., Eds.; European Language Resources Association (ELRA): Luxemburg, 2012; pp. 3562–3567.
68. Pérez, J.M.; Giudici, J.C.; Luque, F. pysentimiento: A Python Toolkit for Sentiment Analysis and SocialNLP tasks. *arXiv* **2021**, arXiv:2106.09462.
69. Prabhat, A.; Khullar, V. Sentiment classification on big data using Naive Bayes and logistic regression. In Proceedings of the 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 5–7 January 2017; pp. 1–5.
70. Lewis, D.D. Naive (Bayes) at forty: The independence assumption in information retrieval. In Proceedings of the European Conference on Machine Learning, Chemnitz, Germany, 21–23 April 1998; Springer: Berlin/Heidelberg, Germany, 1998; pp. 4–15.
71. Domingos, P.; Pazzani, M. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Mach. Learn.* **1997**, *29*, 103–130. [CrossRef]
72. Colas, F.; Brazdil, P. Comparison of SVM and some older classification algorithms in text classification tasks. In Proceedings of the Artificial Intelligence in Theory and Practice: IFIP 19th World Computer Congress, TC 12: IFIP AI 2006 Stream, Santiago, Chile, 21–24 August 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 169–178.
73. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, e00938. [CrossRef] [PubMed]
74. Guo, M.H.; Xu, T.X.; Liu, J.J.; Liu, Z.N.; Jiang, P.T.; Mu, T.J.; Zhang, S.H.; Martin, R.R.; Cheng, M.M.; Hu, S.M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
75. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
76. Stewart, G.W. On the early history of the singular value decomposition. *SIAM Rev.* **1993**, *35*, 551–566. [CrossRef]