LUISS 'Guido Carli'
MSc in Data Science and Management - Data Science in Action Course
May 5, 2024

Ferragni-Balocco Case

A social media-focused perspective on public perception

Project Technical Report

Coci, Marco, Team Leader
ID: 786471, email: m.coci@studenti.luiss.it

Marchioni, Gian Lorenzo
ID: 788811, email: gianlorenz.marchioni@studenti.luiss.it

Paquette, David
ID: 789331, email: d.paquette@studenti.luiss.it

Collaborative Business Case Proposed by Deloitte

1 Introduction

In December of 2022, Italian influencer Chiarra Ferragni took to Instagram to promote a Ferragni-designed Balocco pandoro. The influencer advertised the pandoro to her following of over 29 million, stating that revenues generated from the pandoro sales would be donated to the Regina Margherita Hospital in Turin for bone cancer research. The Ferragni-Balocco pandoro retailed for over 9 euros, while a normal Balocco pandoro retailed for 3.70 euros. In the end, it was discovered that even though the campaign has raised over one million euros, Balocco had made a one-time donation of 50,000 euros prior to the commencement of the collaboration with the influencer, and none afterwards. Customers felt duped, and Ferragni and Balocco were fined by Italy's anti-trust agency (AGCM)[1].

However, that was only the tip of the iceberg for Ferragni and her collaborators, as the once beloved influencer is now being investigated for fraud due to her involvement in the aforementioned campaign, as well as for her involvement in an Easter egg campaign with Dolci Preziozi, a doll campaign with toy maker Trudy, and even a possiby due to her involvement with Oreo biscuits.

As a result of this scandal, this study will aim to understand the public's sentiment on the influencer and how it has shifted over the development of the scandal, as well as to understand repercussions from a financial and communications point of view.

This study will attempt to compare the sentiments between each other, in order to analyze differences in opinions between various sources. All this will be done by analyzing user-generated content on various social media platforms, as well as newspaper articles discussing the scandal and its consequences on her influence and appeal.

2 Methods

2.1 Data Collection

Data was collected through various methods. Due to the Cambridge Analytica scandal, whereby the personal data of millions of Facebook users were used without consent by the British consulting firm Cambridge Analytica, social media platforms greatly restricted access to their APIs[2]. Furthermore, such tight API regulations facilitate compliance with data protection regulations such as GDPR. As such, alternative methods needed to be explored.

To circumvent these limitations, one solution that was discovered was scraping data via a web scraping app called 'Browserflow'[3]. This app enabled the scraping of various posts, as well as the comments, likes, and dates associated with said posts. The app did not come without limitations however, as the free version was only able to scrape comments for a maximum duration of one minute, and could not be used on all platforms (e.g., Facebook); however, approximately 20,000 comments were scraped, with roughly 17,000 of them coming from 98 Instagram posts, and the remaining from X. These Instagram posts were collected from her page, from the pages of various tabloids and newspapers, and from various Instagram pages commenting on the situation, while also collecting the respective comments of each aforementioned post. The X posts were also collected in a similar manner, while filtering for popularity and relevancy. The posts and pages were found mainly by searching for key words such as 'Chiara Ferragni', 'Balocco', and 'Pandoro-Gate'.

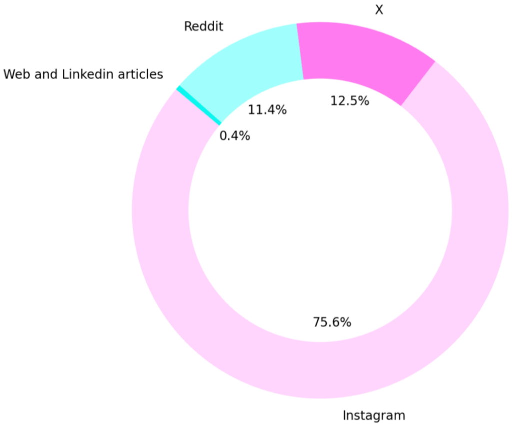

Figure 1: Percentage of source for items in the dataset

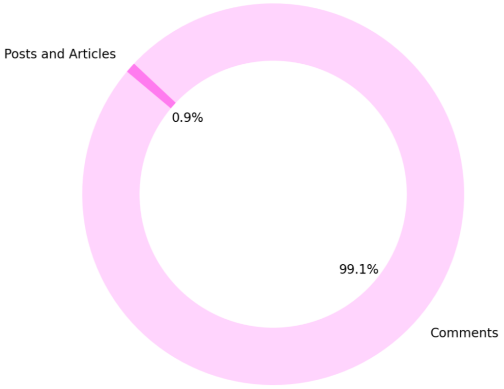

Figure 2: Percentage of text item type. As expected from the extensive social media scraping, the overwhelming majority of items are user comments

Conversly, Reddit's API proved useful for data collection. From Reddit, approximately 2,500 comments were scraped through the use of the API and through the python module PRAW(Python Reddit API Wrapper).

Moreover, newspaper articles pertaining to the scandal and the fallout were also collected. The first way in which these articles were collected was through the use of a Google Custom Search Engine, filtering for the aforementioned words. Through this custom engine, a Google API key, and the BeautifulSoup and requests module on python, the content of 23 articles were scraped. The second way in which articles were scraped was through 'CrawlBase', a webscraping tool which offered a Python compatible library and a crawling API. With this tool, the contents of roughly 100 articles and their respective details (e.g., url, date created, etc.) were scraped. Lastly, through the use of Browserflow, three articles were scraped off of LinkedIn. The remaining articles mainly originated from newspapers of Italian origins, such as '*Il Messaggero*' and '*Il Sole 24 Ore*', with the exception of small amount that were taken from American and British papers such as '*The New York Times*' and '*The Guardian*'.

The data was all collected following a similar structure so that once in the `.csv` file, said files would be easy to merge, and data sources could be easily identified. During the collection process, thorough attention was put into tracking the date for each piece of scraped content:

- X and Reddit allowed for easy retrieval of dates, respectively through selecting the 'date' field of posts and comments during scraping, or simply as feature of the API.
- Scraped web articles required either looking up the creation date by examining the HTML page using BeutifulSoup or manual imputation.
- Instagram comments required converting the format used for comments, which is their age in weeks (e.g. '7w') into an actual date. This method has limited precision, a factor that was kept in mind when plotting time series, for which data was grouped either weekly or monthly.

In figure 3 the time distribution of our gathered content is displayed. Despite trying to uniformly sample the time period from December of 2023 to March of 2024, it is clear that the posts with most engagement usually follow key events.

The variables under which the details of the scraped social media posts and comments, and under which the details of the articles would be classified are:

- `text_content`: The text content of each article, post, and comment scraped.
- `url`: The url of each article, and post scraped, as well as the url of the post under which the scraped comment was made.
- `post_creator`: The creator of each article, post, and comment scraped.
- `date`: The date each article, post, or comment scraped was published.
- `source_category`: The website where each article, post, and comment was scraped from (e.g., Instagram, X, Linkedin, Reddit, etc.).
- `flag`: The classification of the scraped item (e.g., news, opinion, etc.).
- `type_category`: The nature of each scraped item (i.e., post, comment, or article).
- `likes`: How many likes the post or comment received. Exclusively available for Instagram and Reddit items.

In addition to the scraped data, data relating to Chiara Ferragni's Instagram page and her business page, Chiara Ferragni Brand, was collected via InsTrack, an online analytical tool for Instagram. This data includes variables relating to her total followers, the number of users she follows, the number of posts she has made, and the engagement data of her Instagram accounts. This data, along with key events selected from the articles, will compliment the sentiment analysis done with the aforementioned scraped data, and will help better analyse and understand the socioeconomic consequences of the scandal.

2.2 NLP Preprocessing

Once the data had been accumulated and merged together, the dataset had a shape of `(22320, 8)`. The preprocessing then commenced by first targeting the `null` values. The only `null` values were found under the `post_creator`, `likes`, and `date` variables. Since these variables didn't negatively affect the sentiment analysis, the scraped items with `null` values were kept; however, when analysis the dataset, it was discovered that 54 scraped items from Reddit were deleted due to the scraped comment being '`[deleted]`'. These comments were filtered out of the dataset, thus leaving **22,266** scraped items to conduct a sentiment analysis with.

Due to Ferragni's popularity and the gravity of the scandal, social media users from all over the world commented on the matter or posted about it. Furthermore, articles from all major newspapers were discussing it. As such, the scraped dataset was composed of comments in various languages. In order to conduct a proper sentiment analysis, it was necessary to identify the language of each scraped item. As such, an XLM-RoBERTa language detection model from HuggingFace, which had achieved a 0.9977 accuracy on the validation set

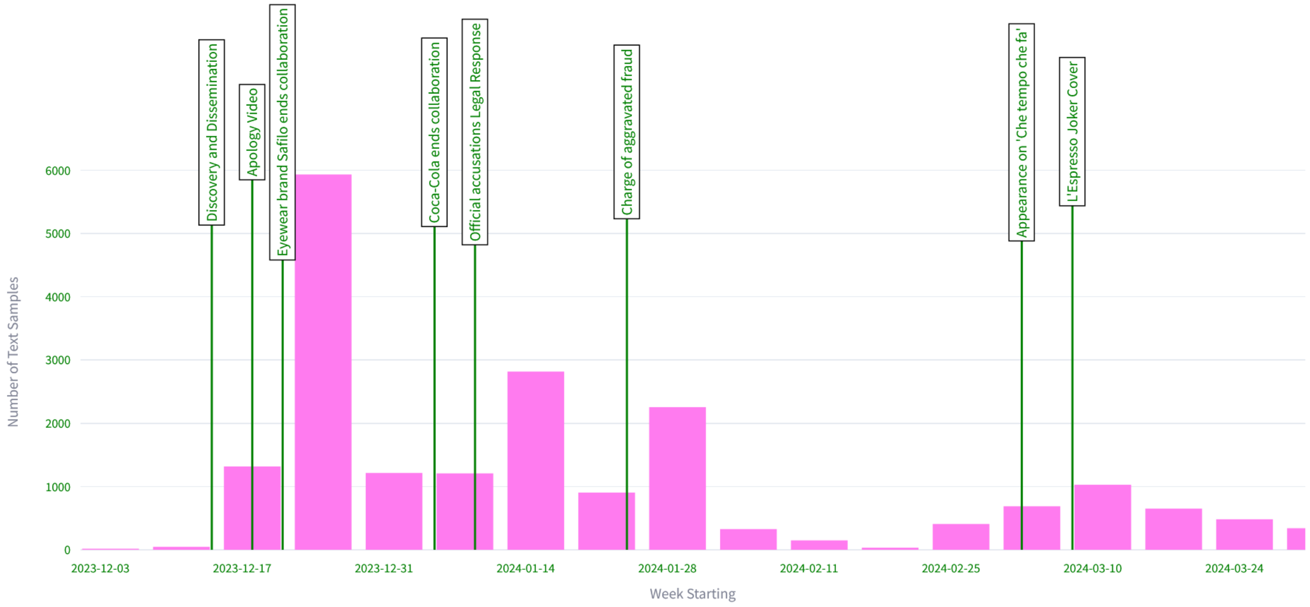

Figure 3: A time distribution of scraped content, grouped weekly. Key events are also listed in table 1

it had been trained on, was applied [4].The model determined that nearly 85% of the dataset was Italian. When analysing, some of the other scraped items in languages other than Italian, it was revealed that, apart from the English items, most languages were wrongly assigned. As such, for simplicity, all non-English comments were assigned Italian. In the end, the dataset was composed of nearly 98% Italian items.

Moreover, due to the fact that most items in the dataset were sourced from various social medias, emojis were often very common; however, as emojis are a concise way of conveying emotions, they could prove to be quite useful in the creation of the final sentiment analysis model. Through the use of the emoji library, the emojis were translated into their respective description. Furthermore, whilst emojis convey emotions, they don't add to the grammatical structure of a sentence. As such, in order to better distinguish the text from the emojis, their descriptions were concatenated to the end of the sentence[5]. A separate dataset was also prepared with emojis being fully removed, in order to conduct a sentiment analysis without emojis, and to juxtapose the results with the ones of the sentiment analysis that has emojis in the model. Therefore, there were now two datasets, both with a shape of `(22266, 8)`

Once the emojis had been either removed or transformed into their description in the scraped items' respective language, it was time to move onto the necessary steps to prepare a text to be put through a sentiment analysis model. These steps can be broken down into major parts: removing non-alphabetic characters, converting uppercase letters into lowercase, tokenizing the texts, removing stopwords and domain words, and lastly lemmatizing each individual word from the tokenized texts. The removal of non-alphabetic characters was done by stripping said non-alphabetic characters. The datasets were then checked for any instances in the `text_content` column that were left empty, or simply full of white spaces. Only 12 rows were removed from the dataset where emojis were transformed into their description, while **1,241** where removed from the emoji-less dataset, most likely due to the fact that these scraped items were solely emojis.

After stripping the scraped items of non-alphabetic characters, all items were then converted to lowercase in order to ensure conformity, and in order to avoid the duplication of words due to uppercase letters. The text was then tokenized, i.e., broken down into a list of individual words, in order to remove the stop words and domain words, and in order to lemmatize the words. The stopwords used were download from the `nltk` library, in both English and Italian. The domain words used included **chiara, ferragni, balocco, pandoro, italian, italiana,** and **influencer**. These aforementioned stop and domain words were used to remove frequent words from the text that held little weight for the sentiment analysis, thus reducing the noise and allowing the model to focus on the more significant words.

The last step of the preprocessing was the lemmatize the preprocessed text, which was also done with the nltk library. Lemmatization is the process of "reducing morphological variants [of words] to one dictionary base form"[6]. For example, the words 'Dancing', 'Dancer', 'Danced', and 'Dances' would all be reduced to the word

'Dance'. On the other hand, stemming, the alternative to lemmatization, "strips affixes from words"[6] in order to obtain its stemmed form. To use the previous example, the stemmed form of 'Dancing', 'Dancer', 'Danced' and 'Dances', would be 'Danc'. The decision to use lemmatization instead of stemming was taken due to the fact that lemmatization produces linguistically correct words, which should theoretically produce more accurate results than stemming; however, this increased accuracy does have its drawbacks, as using a lemmatizer is lengthier, and more computationally expensive process than stemming.

Once the words had been lemmatized, the two dataset were further divided into datasets for their respective language, and the lemmatized words were rejoined together in a new column titled `sentence` in order to use the tokenizer from the pre-trained models from huggingface. These were ultimately the last two steps of of the preprocessing process. After said process, the datasets had 13 columns, 5 more than in the original datasets. These columns were added after every data processing step and included:

- `language`: The language, either English or Italian, of the scraped item.
- `tokenized`: The scraped item in tokenized form.
- `filtered`: The scraped item with domain and stop words filtered out.
- `lemmatized`: The lemmatized scraped item.
- `sentence`: The scraped item rejoined together in a sentence.

As such, the preprocessing resulted in the creation of three more datasets, thus totalling four datasets to be used for the sentiment analysis. They are as following:

- `it`: Dataset of scraped Italian items **with** emojis. It has a shape of **(21874, 13)**.
- `it_ne`: Dataset of scraped Italian items **without** emojis. It has a shape of **(20645, 13)**.
- `en`: Dataset of scraped English items **with** emojis. It has a shape of **(380, 13)**.
- `en_ne`: Dataset of scraped Italian items **without** emojis. It has a shape of **(380, 13)**.

2.3 Sentiment & Emotion Analysis

In order to conduct a sentiment analysis, two different models were sourced from HuggingFace, one for every language. The Italian sentiment analysis model was titled "Italian Bert Sentiment Analysis" [7]. It was created by an Italian start-up called Neuraly, and was created by training the model with a previous iteration of the model that was created with a dataset of Italian Wikipedia pages, and then fine-tuned on a dataset of Tweets. As hyperparameters, the model used an `AdamW` optimizer, which controls weight decay in order to prevent overfitting [8]. The model also set `max epochs` to 5, `batch size` to 32, and enabled `early stopping with patience` at 1 to prevent overfitting. This model achieved an 82% accuracy on its test set. Similarly, the English sentiment analysis model used , titled "Twitter-roBERTa-base for Sentiment Analysis"[9], was also a variant of a BERT model that was trained on a Twitter dataset titled "tweet_eval". While no information was found on the hyperparameters set for the model, the Github page of the creator stated that the hyperparameters were fine-tuned with a Python library titled "Ray Tune". Moreover, while the model did not explicitly state the accuracy achieved, the previous iteration of said model achieved both an F1-Score and an accuracy of 71% on the test set. Both these models outputted the probabilities that an item belong to one of three following possible sentiments: `negative, neutral`, or `positive`.

Additionally, in order to compliment the sentiment analysis, another analysis was performed on each language dataset which focused on the emotions expressed in scraped items, rather than on the sentiments said items displayed. For the Italian datasets, the model was titled "FEEL-IT: Emotion and Sentiment Classification for the Italian Language"[10], a fine-tuned umBERTo model trained on a dataset of tweets, which obtained an accuracy of 73%. As an output, this model gave the probabilities that the item belonged to one of the four following emotions: `anger, joy, fear`, and `sadness`. On the other hand, the English emotion model, titled "Emotion English DistilRoBERTa-base"[11], was a DistilRoBERTa model trained on a dataset of various text types all sourced from various mediums(i.e., Twitter, Reddit, student self-reports, and TV dialogues). The model achieved a 66% accuracy on its test set, and outputs the probability that a text belongs to each of the following emotions: `joy, anger, disgust, fear, sadness, neutral`, and `surprise`; however, slight modifications were done to the parameters of this model due to the fact that the length of certain articles were too long for it, and as such, the parameter `max_length` had to be set to 512 tokens.

2.4 BERT Model

One important thing to note, is that all models mentioned within this paper, that is, the model used for language classification, and the models used for the sentiment and emotion analysis are all variations of a BERT model, an open source machine learning framework for natural language processing made public by Google in 2018 [12]. BERT is an acronym for Bidirectional Encoder Representations from Transformer. The reason it is has the title of "Bidirectional" is due to the fact that it uses a bidirectional self-attention mechanism which enables it to look at the context of each word from the left and right side of the sentence, thus enabling it to obtain more accurate results as it has more contextual understanding [13]. The variations of BERT models used throughout this project are:

- BERT: The original version of the BERT model. Only used for the Italian Sentiment Analysis.
- roBERTa: Robustly Optimized BERT Approach. Is a version of BERT that is trained more extensively on larger datasets with fine-tuned hyperparameters. As such, it is more computationally expensive. This type of model is used for the language detection, and the English sentiment model. [14]
- umBERTo: BERT model adapted for the Italian language. Only used for the Italian emotions analysis model.[15]
- DistilRoBERTa: A faster, smaller version of a roBERTa model. Only used for the English emotions analysis model. [16]

3 Results and Discussion

Once the results were obtained from the sentiment and emotional analysis in both languages, the results were concatenated into one large dataset, and subsequently separated into various datasets in order to fully compare the sentiments and emotions from every source.The datasets are as follow:

- final_data and final_data_ne: The large results dataset with the results from all four models, for the data with emojis and without emojis.
- insta and insta_ne: The dataset comprising of the results for only the items scraped from Instagram, with and without emojis.
- X and X_ne: The dataset comprising of the results for only the items scraped from X(previously Twitter), with and without emojis.
- reddit and reddit_ne: The dataset comprising of the results for only the items scraped from Reddit, with and without emojis.
- article and article_ne: The dataset comprising of the results for only the items scraped from articles, with and without emojis.
- cf_insta and cf_insta_ne: The dataset comprising of the results for only the items scraped from Chiara Ferragnis' Instagram page, with and without emojis.
- cfb_insta and cfb_insta_ne: The dataset comprising of the results for only the items scraped from the Chiara Ferragni Brand Instagram page, with and without emojis.
- insta_no_cf_insta and insta_no_cf_insta_ne: The dataset comprising of the results for the items scraped from Instagram, excluding those scraped from Chiara Ferragnis' Instagram page and from Chiara Ferragni Brands' Instagram page, with and without emojis.
- it and it_ne: The dataset comprising of the results for only the Italian scraped items, with and without emojis.
- en and en_ne: The dataset comprising of the results for only the English scraped items, with and without emojis.

In order to compare the results, the mean of every probability for every sentiment and emotion was taken, as well as the count of the sentiment and emotion which had the highest probability for every text. Please refer to 4 in order to see the results for all the specific datasets.

Overall, the highest sentiment across the entire dataset, apart from the English dataset, was neutral. The largest mean value was for the neutral sentiment was found in the article dataset, which was to be expected as articles are inherently neutral. While the positive sentiment had a slightly higher mean in the English dataset, the neutral sentiment still had a higher max count, suggesting that certain articles may have been overwhelmingly positive, thus increasing the mean positive sentiment value.

The second highest mean probability value was for the negative sentiment. Most mean values for this sentiment ranged from 0.25 to 0.33. The most negative source of items were taken from Instagram when excluding any post from Chiara Ferragnis' Instagram page or from her brand page without any emojis, with a mean negative sentiment score of 0.327, followed by items scraped from Reddit without emojis, with a mean negative sentiment score of 0.320; however, Instagram had much more instances where the max sentiment was negative, perhaps also due in part to the sheer number of scraped Instagram items.

Content Category	**Mean Sentiment Scores**			**Sentiment Counts**		
	Positive	**Neutral**	**Negative**	**Positive**	**Neutral**	**Negative**
Total	0.21	0.49	0.30	4569	11034	6651
Total w/o Emojis	0.16	0.53	0.31	3228	11308	6489
Instagram	0.24	0.45	0.31	3914	7854	5066
Instagram w/o Emojis	0.17	0.51	0.32	2584	8117	4906
CF Instagram	0.39	0.41	0.20	1373	1465	718
CF Instagram w/o Emojis	0.33	0.44	0.23	994	1326	701
CF Brand Instagram	0.30	0.44	0.27	342	277	142
CF Brand Instagram w/o Emojis	0.23	0.45	0.32	202	228	196
Instagram w/o CF Posts	0.19	0.48	0.33	2313	6047	4146
Instagram w/o CF Posts w/o Emojis	0.13	0.54	0.33	1448	6514	4009
X	0.14	0.59	0.27	351	1681	747
X w/o emojis	0.14	0.59	0.27	351	1681	747
Reddit	0.12	0.56	0.32	300	815	1426
Reddit w/o Emojis	0.01	0.56	0.32	288	813	1438
Articles	0.06	0.66	0.28	4	22	71
Articles w/o Emojis	0.06	0.66	0.28	5	22	70
Italian	0.15	0.53	0.31	4425	11064	6585
Italian w/o Emojis	0.16	0.53	0.31	3096	11126	6423
English	0.39	0.42	0.22	144	170	66
English w/o Emojis	0.36	0.42	0.22	132	182	66

Figure 4: Sentiment Analysis Results

As alluded to in the previous paragraph, the removal of emojis seemed to solely increase the mean `negative` sentiment and decrease the mean `positive` sentiment of every item from every source, excluding X's in which all sentiments remained constant, suggesting a lack of emojis in the originally scraped items, and excluding the items scraped from articles, where the removal of emojis actually slightly increased the mean `positive` sentiment. Not only did it affect the mean sentiment value, but also the count of highest sentiments, boosting the count of `negative` sentiments increase for the entire dataset, and decreasing that of the `positive` sentiment. The largest jump in mean `negative` sentiment was between Instagram items scraped from her brands' page, where the mean `negative` sentiment jumped from `0.265` with emojis, to `0.320` without. A possible hypothesis for this jump in negativity is the fact that certain social media users, particularly on Instagram, comment solely in emojis. Hypothetically, if someone were to comment solely laughing emojis, due to the preprocessing done where the emojis were transformed into their description, that comment could be taken as positive due to the fact that that emoji's description is "face with tears of joy", regardless of the fact that that emoji could have been used to symbolize the user ridiculing the person in question, or using it sarcastically. As such, a dataset without emojis negates that possible issue, and thus would logically increase the mean `negative` sentiment.

An interesting juxtaposition is between the mean values of the sentiments between the items scraped from Chiara Ferragnis' Instagram page, with and without emojis, and from the rest of Instagram, excluding any item collected of her or her brands' page. In this case, the results illustrate a jump in `0.12` for the mean `negative` sentiment for the data with emojis (figure 5), and one of `0.10` for the data without emojis (6). Also, as highlighted in figures 8 and , her engagement on Instagram over the course of this scandal was much lower than her normal rate. This is mainly due to the fact that she limited comments, and was also more than likely deleting any negative comment in order to mitigate the damage.

When analysing the results of the emotional analysis, the predominant emotion was `anger`, with mean probability values reaching as high as `0.71` in items scraped from articles. The second highest emotion was `joy`, which unsurprisingly came from the items scraped from her and her brands' Instagram pages, as she had limited the negativity as previously mentioned.

Also in similar fashion to what was previously said about emojis, the removal of the emojis only amplified the `anger` emotion, and reduced the mean value of the `joy` emotion. The largest change in these emotions also came about in items scraped from Ferragnis' Instagram page, with the mean `joy` value dropping nearly `0.18` points, from `0.527` with emojis to `0.352` without them, and the mean value of `anger` increasing from `0.284` with them, to `0.351`. This drop in `joy` is most likely due to similar reason previously mentioned, coupled with the fact that her supporters may have also solely commented positive emojis which were consequently removed.

3.1 Business Insights

Through comprehensive research, a series of pivotal events were identified that critically influenced the business dynamics and public image of Chiara Ferragni. These events signify a progressive escalation that reshaped her professional landscape.

Date	**Event**
05/12/22	Launch of the Campaign
14/12/23	Discovery and Disclosure
18/12/23	Apology Video and Donation
21/12/23	Eyewear brand Safilo ends collaboration
04/01/24	Coca-Cola ends collaboration
09/01/24	Official accusations and Legal Response
24/01/24	Charge of aggravated fraud
03/03/24	Appearance on 'Che Tempo Che Fa'
08/03/24	'*L'Espresso*' Joker cover

Table 1: Timeline of key Events

Each incident significantly deepened the impact on Ferragni's business commitments and her standing in the eyes of the public. The revelation of undisclosed details on December 14, 2023 was quickly followed by remedial efforts, including a public apology and charitable donations.

The immediate consequences were serious. Major brands such as Safilo and Coca-Cola ended their partnerships in early January 2024, indicating a serious breach of trust, thus leading to a re-evaluation of her suitability as brand ambassador. In March of 2024, Ferragni faced ongoing legal challenges and increased media scrutiny, highlighted by her portrayal in the magazine "L'Espresso".

Naturally, other brands associated with Ferragni were also mentioned in the dataset. Figure 7 displays the number of instances of the 10 most frequently mentioned were named in the dataset.

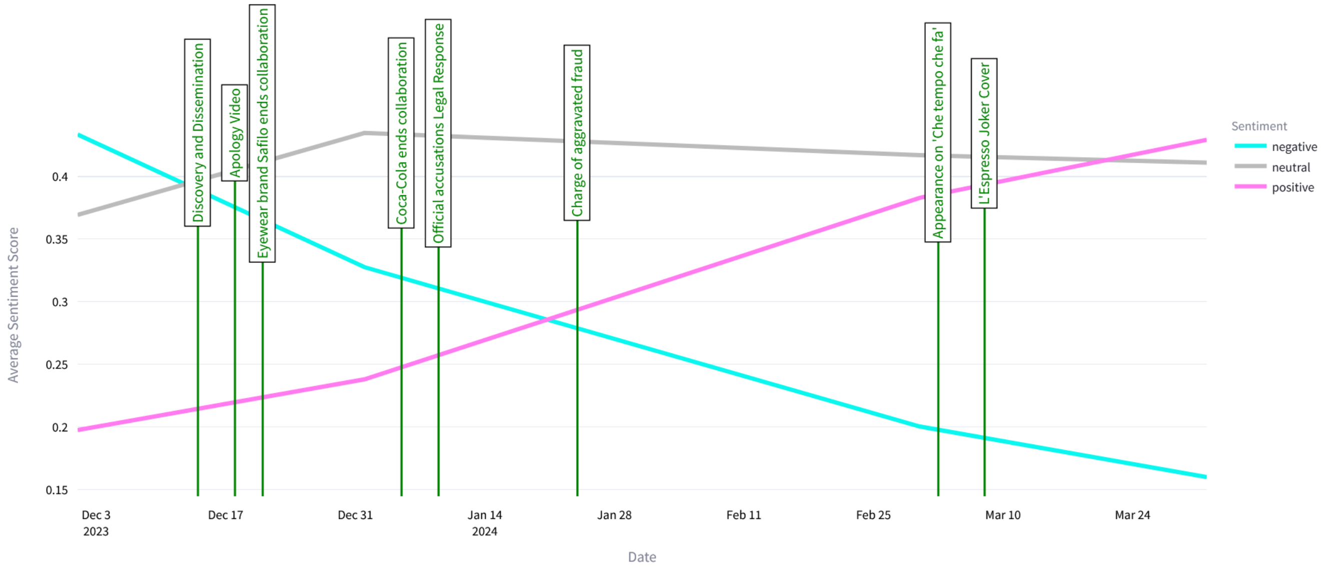

Figure 5: Sentiment trend for CF's Instagram Profile, With Emojis

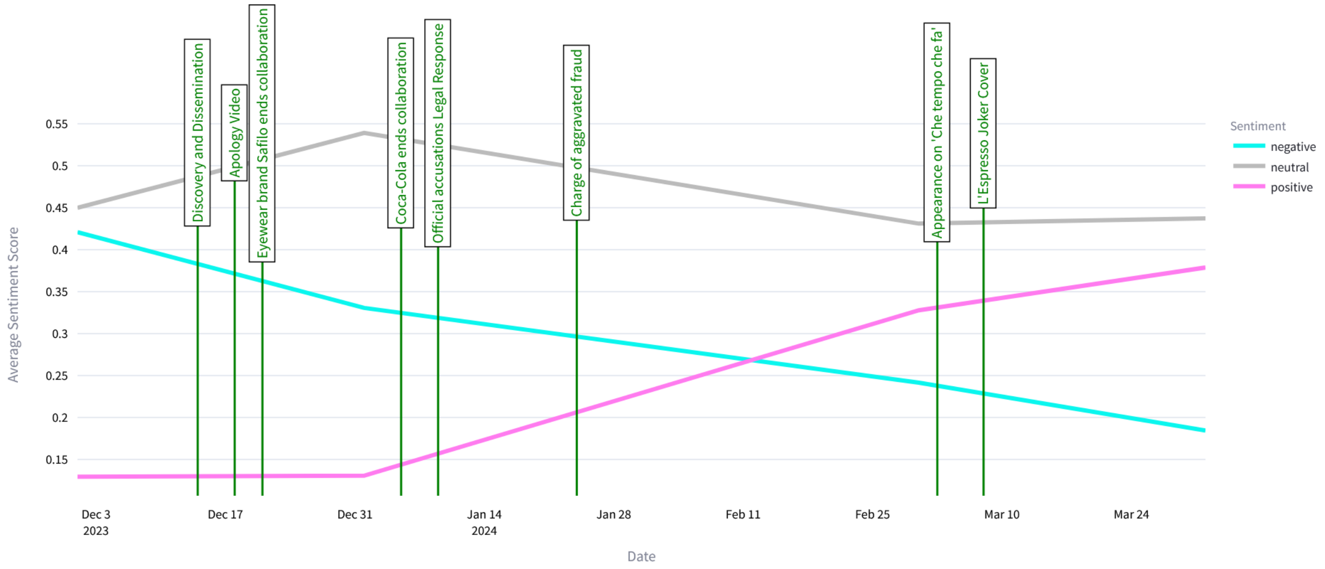

Figure 6: Sentiment trend for CF's Instagram Profile, Without Emojis

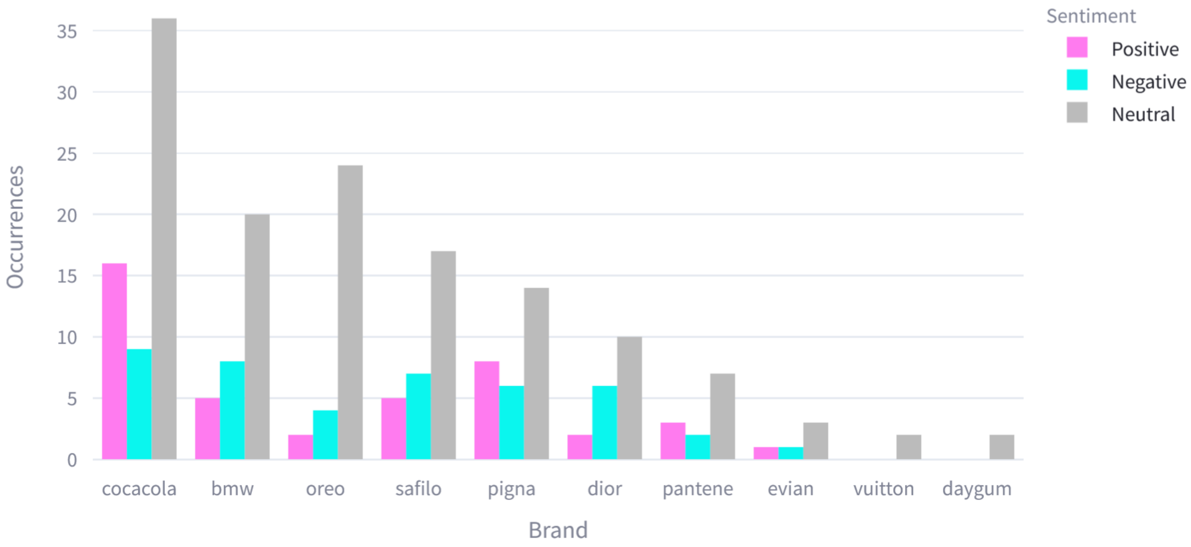

Figure 7: Brand perception sentiment analysis

Social media analysis provided quantifiable information on the repercussions of the scandal. Ferragni's Instagram follower count plummeted from nearly 29.7 million to less than 29 million. As seen in the following graph, the event are in the table 1 .

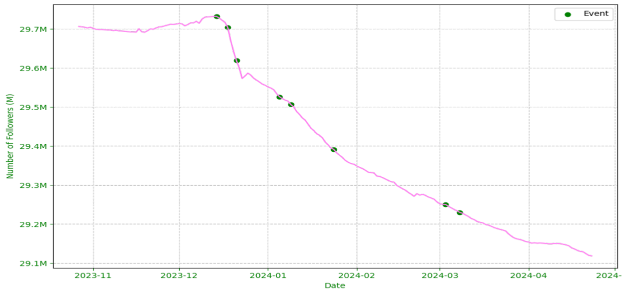

Figure 8: Follower Counts over time for CF's Main Instagram Profile

Initially, the frequency of her posts increased, perhaps in an attempt to manage public perception, and mitigate damage with an apology video; however, this number dropped significantly as the scandal unfolded, indicating a strategic retreat from public engagement and a substantial loss of potential earnings, as her endorsements were valued at around €93,000 each. If this number holds true, Chiara Ferragni may have already accrued losses of over **five million euros** [17].

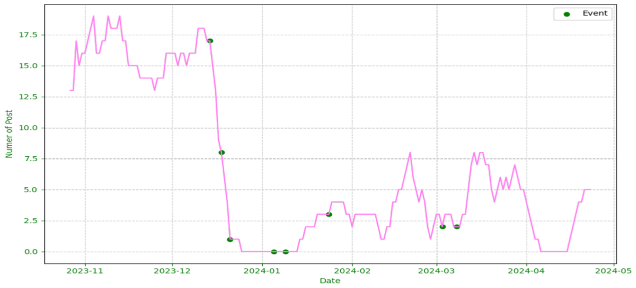

Figure 9: Posts count over time for CF's main Instagram Profile

Public engagement peaked during the initial disclosure, but dropped dramatically after Ferragni began blocking comments on her posts, underscoring the intense scrutiny and controversy surrounding the allegations. This was followed by a notable decrease in engagement, suggesting a reduction in public interest, or perhaps in an alternative method of voicing said interest.

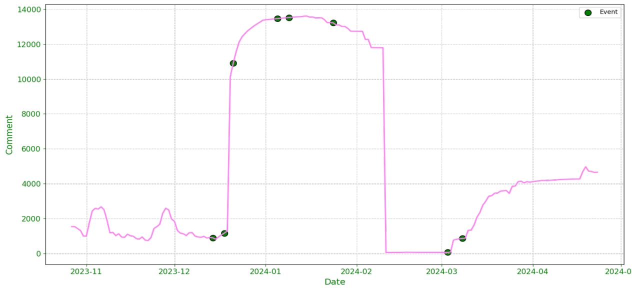

Figure 10: Comment Count over time for CF's Main Instagram Profile

Furthermore, the impact on the Ferragnis' fashion brand, which retails in around 320 stores, has also been considerable. While specific financial details were not disclosed, the brand's Instagram profile also suffered a significant loss of followers, further exacerbating the challenges faced following the scandal.

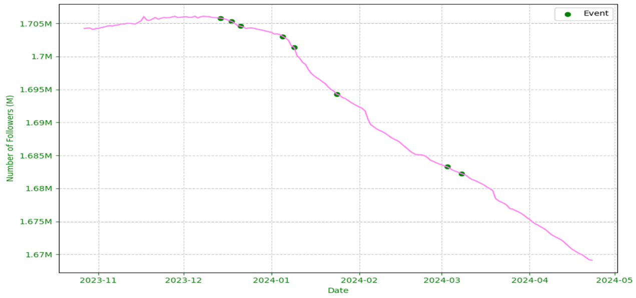

Figure 11: Follower count over time for CF's Brand Instagram Profile

4 Conclusions

This reportage meticulously analyzed the repercussions of the Pandoro-Gate scandal on Chiara Ferragni's career, highlighting its profound effects on her public image and her entrepreneurial activities. The results of the sentiment analysis, derived from data collected from social media platforms and newspaper articles, revealed a worsening trajectory of public sentiment, reflecting a growing disenchantment with Ferragni among the public.

There has been a marked decline in social media interaction and a loss of followers for Ferragni, exacerbated by the events listed above.

The termination of partnerships with major brands such as Safilo and Coca-Cola highlights the significant impact on its commercial credibility and brand trust.

Sentiment analysis showed a shift from neutral to increasingly negative, indicating a more critical public reaction towards Ferragni as the scandal unfolded.

Another thing to notice is that although she tried to "improve" her situation by going on "*Che tempo che fa*"' or via other public appearances, the negative sentiment increasingly negative over time so your continued to increase over time, suggesting failures in her damage limitation strategies.

The financial quantification of the impact of the scandal remains uncertain. While the study notes the loss of followers and partnerships, a more detailed assessment is needed to directly correlate these losses with the financial impacts we failed to make. The broader effects on the influencer marketing industry have been addressed, but a detailed investigation into how similar scandals could reshape industry standards and regulatory measures would be useful. Future research could benefit from a longitudinal study that tracks patterns of recovery in social media engagement and public sentiment over time following the scandal. Comparative studies with other similar incidents could provide deeper insights into the dynamics of public trust and recovery in digital influencers' careers.

To conclude, this investigation highlights the imperative need for transparency and ethical practices in influencer marketing, particularly when charitable promises are involved. As the digital landscape evolves, the frameworks that govern these new forms of celebrity and influence must consequently evolve, ensuring accountability and fostering a sustainable environment for future digital marketing practices.

Appendix A: Code Description

The code commences by importing the specific libraries necessary. A total of 15 libraries were imported, including `Pandas`, `emojis`, and modules from the `Transformers` library for preprocessing. `PyTorch` and modules from the `transformers` library were used for the models, and lastly, `Plotly` library was used for visualisations.

The first function created was done in order to predict the language of every scraped item, as it was necessary to do so in order to properly conduct a sentiment analysis. It was done using a pipeline from Transformers.

Algorithm 1 NLP Predict Language Pipeline

```
LOAD transformers.pipeline with xlm-roberta-base-language-detection as model
Function predict_language(text)
  CALL transformers.pipeline(text)
  RETURN language label string
Apply predict_language(text) to dataframe
```

Once the function was applied to the dataset, and the language of each text was identifiable in the 'language' column of the dataset, all languages besides English are converted to Italian. This was done to simplify the final model, as well as due to the fact that when analysing the outputted languages, most texts were in Italian, rather than in the predicted languages.

Since emojis can carry many sentiments and emotions, in order to not simply remove them, the following function was then used to process the emojis:

Algorithm 2 Emoji Description Concatenation

```
Function emoji2concat_description
  Initialize emoji_list to get description of each emoji text
  Strip text of emojis
  Iterate through list of emojis to obtain emoji description in specific language
  Concatenate description to end of text
  RETURN new text with description
FOR each row in dataframe
  Get language and text from each row
  Apply function to row
```

Additionally, a separate dataset was created where the emojis all stripped and replaced with empty spaces in order for comparison.

The dataset was then put through typical NLP preprocessing tasks, i.e., removing non-alphabetic characters, converting all letters to lowercase, tokenizing the texts, removing the stop and domain words, and lemmatizing the tokenized words. Modules from the `nltk` library were used for these aforementioned processes.

In order to properly conduct a sentiment and emotional analysis, the datasets was separated based on the languages of the texts. The lemmatized words were also reunited with spaces in order to use the tokenizers of the pretrained models in the column '`sentence`'.The first sentiment analysis was completed on the Italian dataset as follows:

Algorithm 3 Italian Sentiment Analysis

```
LOAD AutoTokenizer for pretrained model neuraly/bert-base-italian-cased-sentiment
LOAD AutoModelForSequenceClassification for "neuraly/bert-base-italian-cased-sentiment"
Initialize device based on CPU
Function italian_sentiment(sentence)
    TOKENIZE sentence using AutoTokenizer
    MOVE tokenized sentence to device
    PREDICT using model
    CONVERT predictions to probabilities using softmax
    CREATE dictionary with sentiment label (negative, neutral, or positive) as key and probability score as value
    RETURN dictionary
FOR each sentence in Italian Dataset:
    Apply italian_sentiment function
    Store results in 'sentiment' column
FOR each sentence in Italian Dataset Without Emojis:
    Apply italian_sentiment function
    Store results in 'sentiment' column
```

An analysis of the emotions was subsequently undertaken for the Italian datasets:

Algorithm 4 Italian Emotion Analysis

```
LOAD classifier for text classification from "MilaNLProc/feel-it-italian-emotion"
Function italian_emotion(sentence)
    PREDICT using classifier
    RETURN prediction
Initiate empty columns in Italian datasets titled 'emotion'
FOR each sentence in Italian Dataset:
    Apply italian_emotion function
    Store results in 'emotion' column
FOR each sentence in Italian Dataset Without Emojis:
    Apply italian_emotion function
    Store results in 'emotion' column
```

Similar processes for sentiment and emotion analysis were undertaken for the English dataset.

Algorithm 5 English Sentiment Analysis

```
LOAD AutoTokenizer, AutoModelForSequenceClassification, and AutoConfig from "cardiffnlp/twitter-roberta-base-sentiment-latest"
Function english_sentiment(sentence)
    TOKENIZE sentence using tokenizer
    COMPUTE output using model
    EXTRACT scores from output
    CONVERT scores to probabilities using softmax
    CREATE dictionary with sentiment label (negative, neutral, or positive) as key and probability score as value
    RETURN dictionary
FOR each sentence in English Dataset:
    Apply english_emotion function
    Store results in 'sentiment' column
FOR each sentence in English Dataset Without Emojis:
    Apply english_emotion function
    Store results in 'sentiment' column
```

Algorithm 6 English Emotion Analysis

```
LOAD classifier for text classification from "j-hartmann/emotion-english-distilroberta-base"
Function english_emotion(sentence)
    PREDICT using classifier
    RETURN prediction
Initiate empty columns in English datasets titled 'emotion'
FOR each sentence in English Dataset:
    Apply english_emotion function
    Store results in 'emotion' column
FOR each sentence in English Dataset Without Emojis:
    Apply english_emotion function
    Store results in 'emotion' column
```

Both the Italian and English datasets, with and without emojis were then concatenated to one final dataset; however, the output of the emotion functions came out as a nested list for both the Italian and English functions. As such, in order to ensure consistency, the columns for all four datasets had to be transformed. The following function enabled consistent format with the other analysis columns:

Algorithm 7 Extract and Transform Emotion Columns

```
Function extract_and_transform(nested_list)
    Initialize dictionary for results
    For each list in nested list
        For dictionary in each list
            Extract emotion label to be used as key
            Extract score to be used as value
    RETURN dictionary
Apply function to dataset
```

For analytical reasons, the final dataset was then divided into smaller datasets, in order to compare the sentiments and emotions originating from every source. Furthermore, in order to best compare the sentiments and emotions, the mean of the sentiments and emotions was taken, as well as the count of every instance in which said sentiment or emotion had the highest probability. These calculations were done with the following functions:

Algorithm 8 Calculate mean of sentiments

```
Function sentiment_mean(df, column_name = 'sentiment')
    Initialize dictionary to calculate sum of probabilities per sentiment
    Initialize count
    For dictionary in column
        For key in dictionary
            UPDATE sum
        UPDATE count
    Calculate mean for sentiment in dictionary
    Return dictionary
Apply function to dataset
```

Algorithm 9 Calculate Count of Highest Sentiment

```
Function count_max_sentiment(df, column_name)
    Initialize counter
    For dictionary in column
        SELECT emotion with highest probability
        UPDATE count per emotion
    Return count per emotion as dictionary
Apply function to dataset
```

Algorithm 10 Calculate mean of emotions

```
Function emotion_mean(df, column_name = 'emotion')
  Initialize dictionary to calculate sum of probabilities per emotion
  Initialize count
  For dictionary in column
    For key, value in dictionary
      UPDATE sum
    UPDATE count
  Calculate mean for emotion in dictionary
  Return dictionary
Apply function to dataset
```

Algorithm 11 Calculate Count of Highest Emotion

```
Function count_max_emotion(df, column_name)
  Initialize counter
  For dictionary in column
    GET emotion with highest probability
    UPDATE count per emotion
  Return count per emotion as dictionary
Apply function to dataset
```

These functions were all used to create the final results dataset used for analysis. This dataset was created through the use of two functions in order to automize and optimize the process. The functions are as follows:

Algorithm 12 Process data

```
Function process_data(df, column_name)
  Initialize results dictionary with values calculated by previously defined functions (i.e., sentiment_mean,
count_max_sentiment, emotion_mean, count_max_emotion).
RETURN results dictionary
```

Algorithm 13 Prepare Dataset

```
Function prepare_and_process_data(df, column_name)
  Initialize results dictionary
  WITH ThreadPoolExecutor, submit process_data tasks for each data source
  WAIT for all futures to complete
RETURN results dictionary
```

Appendix B: Author Contribution

Term	Definition	Member
Conceptualization	Ideas; formulation or evolution of overarching research goals and aims	Gian Lorenzo
Methodology	Development or design of methodology; creation of models	David
Software	Programming, software development; designing computer programs; implementation of the computer code and supporting algorithms; testing of existing code components	Marco and David
Validation	Verification, whether as a part of the activity or separate, of the overall replication/ reproducibility of results/experiments and other research outputs	Marco
Formal analysis	Application of statistical, mathematical, computational, or other formal techniques to analyze or synthesize study data	Gian Lorenzo
Investigation	Conducting a research and investigation process, specifically performing the experiments, or data/evidence collection	Gian Lorenzo
Resources	Provision of study materials, reagents, materials, patients, laboratory samples, animals, instrumentation, computing resources, or other analysis tools	Marco
Data Curation	Management activities to annotate (produce metadata), scrub data and maintain research data (including software code, where it is necessary for interpreting the data itself) for initial use and later reuse	Gian Lorenzo
Writing - Original Draft	Preparation, creation and/or presentation of the published work, specifically writing the initial draft (including substantive translation)	David
Writing - Review	Editing Preparation, creation and/or presentation of the published work by those from the original research group, specifically critical review, commentary or revision – including pre-or post-publication stages	David
Visualization	Preparation, creation and/or presentation of the published work, specifically visualization/ data presentation	Marco
Supervision	Oversight and leadership responsibility for the research activity planning and execution, including mentorship external to the core team	Marco
Project administration	Management and coordination responsibility for the research activity planning and execution	Gian Lorenzo
Funding acquisition	Acquisition of the financial support for the project leading to this publication	David

References

[1] Angela Giuffrida. Italian influencer investigated over christmas cake charity scheme. https://www.theguardian.com/world/2024/jan/09/italian-influencer-chiara-ferragni-investigated-over-christmas-cake-charity-scheme, 2024.

[2] Ronnie Mitra. How the facebook api led to the cambridge analytica fiasco. https://apiacademy.co/2018/06/how-the-facebook-api-led-to-the-cambridge-analytica-fiasco/, 2018.

[3] Browserflow - web scraping & web automation. https://browserflow.app/.

[4] A fine-tuned version of xlm-roberta-base on the language identification dataset. https://huggingface.co/papluca/xlm-roberta-base-language-detection.

[5] Bale Chen. Emojis aid social media sentiment analysis: Stop cleaning them out! https://towardsdatascience.com/emojis-aid-social-media-sentiment-analysis-stop-cleaning-them-out-bb32a1e5fc8e, 2023.

[6] Eda Kavlakoglu Jacob Murel Ph.D. What are stemming and lemmatization? https://www.ibm.com/topics/stemming-lemmatization, 2023.

[7] Neuraly. Italian bert sentiment model. https://huggingface.co/neuraly/bert-base-italian-cased-sentiment?text=Huggingface+

[8] Fabio M. Graetz. Why adamw matters. https://towardsdatascience.com/why-adamw-matters-736223f31b5d, 2018.

[9] Twitter-roberta-base for sentiment analysis - updated (2022). https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest, 2022.

[10] Federico Bianchi, Debora Nozza, and Dirk Hovy. Feel-it: Emotion and sentiment classification for the italian language. https://huggingface.co/MilaNLProc/feel-it-italian-emotion?text=Mi+piaci.+Ti+amo, 2021.

[11] Jochen Hartmann. Emotion english distilroberta-base. https://huggingface.co/j-hartmann/emotion-english-distilroberta-base, 2022.

[12] Carmen Hashemi-Pour. Bert language model. https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model, 2024.

[13] Sujit Pal Amita Kapita, Antonio Gulli. Deep learning with tensorflow and kera. Packt Publishing Ltd., 2022.

[14] Drishti Sharma. A gentle introduction to roberta. https://www.analyticsvidhya.com/blog/2022/10/a-gentle-introduction-to-roberta/, 2022.

[15] Loreto Parisi, Simone Francia, and Paolo Magnani. Umberto: an italian language model trained with whole word masking. `https://github.com/musixmatchresearch/umberto`, 2020.

[16] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. `https://huggingface.co/distilbert/distilroberta-base`, 2019.

[17] Il Messagero. Chiara ferragni perde 5 milioni di euro, l'effetto boomerang della strategia social. https://www.ilmessaggero.it/persone/ferragni_effetto_boomerang_persi_5_milioni_euro_strategia_social-7994652.html, 2024.