Inteligencia Artificial 26(71), 114-130
doi: 10.4114/intartif.vol26iss71pp114-130

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Sentiment Gradient - Improving Sentiment Analysis with Entropy Increase

Fernando Cardoso Durier da Silva, Ana Cristina Bicharra Garcia, Sean Wolfgand Matsui Siqueira
Federal University of the State of Rio de Janeiro - UNIRIO
fernando.durier@uniriotec.br
cristina@uniriotec.br
sean@uniriotec.br

Abstract Information sharing on the Web has also led to the rise and spread of fake news. Considering that fake information is generally written to trigger stronger feelings from the readers than simple facts, sentiment analysis has been widely used to detect fake news. Nevertheless, sarcasm, irony, and even jokes use similar written styles, making the distinction between fake and fact harder to catch automatically. We propose a new fake news Classifier that considers a set of language attributes and the gradient of sentiments contained in a message. Sentiment analysis approaches are based on labelling news with a unique value that shrinks the entire message to a single feeling. We take a broader view of a message's sentiment representation, trying to unravel the gradient of sentiments a message may bring. We tested our approach using two datasets containing texts written in Portuguese: a public one and another we created with more up-to-date news scrapped from the Internet. Although we believe our approach is general, we tested for the Portuguese language. Our results show that the sentiment gradient positively impacts the fake news classification performance with statistical significance. The F-Measure reached 94 %, with our approach surpassing available ones (with a p-value less than 0.05 for our results).

Keywords: Fake News, Gradient, Sentiment Analysis, Machine Learning, NLP.

1. Introduction

Taking advantage of the overwhelming amount of information on the Web, big corporations, governmental organizations, and ill-intended people may use technology to spread propaganda, manipulate the

ISSN: 1137-3601 (print), 1988-3064 (on-line)

information people will consume and misguide the beliefs of the masses. Worsening the situation, psychology theories have shown that people prefer fake information even in the presence of the known truth [18]. The reasons for this preference include the low cognitive effort for understanding the fake message, the high social acceptance of the lies, and even the old saying that repeating a lie many times turns it into a truth [23].

From the automatic detection process, we find out that sentiment analysis plays an important role in detecting the urgency feeling that boosts the fake news spread, or the serious tone of a message. Works that use such features rely on discrete values that imply either happiness, neutrality, or sorrow, while when in reality, we can see a varying spectrum of those in a simple single sentence [19].

We developed a model capable of handling the multi-classification aspect of a text and the sentimental variance. We trained and tested on a set of sociopolitical Brazilian Portuguese news articles we outsourced and on a public dataset.

This paper presents our gradient sentiment method for detecting fake news that outperforms current fake news detection approaches. Next section presents the related works that provides an overview of the state-of-the art in fake news detection. Then we present the Sentiment Gradient method followed by the datasets' descriptions and the experiments. Last, we present the discussion of the results reflecting on the reasons our method outperform the others and the conclusion.

2. Related Works

For fake news detection, the literature defines different strategies to solve the problem of labeling data as truth or lie. Shu et al. [28], in an extensive review, pointed that the trend of sentiment analysis in fake news Detection is fundamental for completing this task. Wang et al. [33] analyzed the different feature choices available in this research area, one of them the sentiment analysis of fake news, that they divided into four possible classes (Factual, Manipulative, Hoax, and Incomplete) and with pure sentiment analysis (provided by Linguistic Inquiry and Word Count - LIWC technique) achieved 94.2 % of accuracy for a known English set the PoliFact. Bhutani et al. [3] relied upon the traditional term frequencyâinverse document frequency(TF-IDF) vectorizer modeling of fake news, in order to be compliant to neural networks models, achieving 84.7 % of accuracy.

Monteiro et al. [19] proposed a new Portuguese Dataset with 7200 news (3624 True News and 3576 fake news) called Fake.BR. They tested classifiers upon their dataset and got 89 % for their best classifier, an Support Vector Machine(SVM), for all the features they worked with (POS-Tagging, Word Embedding, Sentiment Analysis, etc.). Focusing only on sentiment analysis, they got the best score of 56 % accuracy.

We observe in the literature that the classical machine learning models are used as baselines of comparison against neural networks models [17], which are modern and less feature engineering oriented approaches (as their inner architecture is capable of automatic feature extraction) [16]. However, we should not forget that the classical models can be as effective as the neural ones; the main difference is that they require more manual effort on feature engineering. Manjusha and Raseek [16] obtained a winning 79.7 % mean F-Measure for classifying articles into satire, humor, and irony with a Convolutional Neural Network(CNN) that competed against SVM, Decision Tree, K-Nearest Neighbors(KNN), and Gaussian Naive Bayes(GNB). On the other hand, de Morais et al. [20] obtained 80 % F-Measure with classical models only for a Brazilian set of Portuguese News.

According to the literature, the datasets are outsourced from social networks and microblogs originated from polemic subjects as elections, polarizing discussions, and events. Few works deal with the Portuguese language, such as: [22]; [8]; [26]; [20]. They all have in common the same outsourcing strategy of web scraping articles from the news portals available in Brazil, and relying upon these portals' reputation to label them as true or fake news. The trend we might notice is that they follow the same strategy of English oriented works, and were implemented almost simultaneously, therefore having no direct reference or intersection with one another. They chose as metrics f-measure, accuracy, precision, and recall. Moreover, they are also based on the classical machine learning models.

The works that deal with sentiment analysis usually focus on a sum of sentiment scores. Luo et al. [15] try to change the perspective for financial news analysis by configuring an Long short-term memory neural net(LSTM) model to have attention over sentences instead of words, but, in the end, it still aggregates the sentiment score. The works of Wang et al. (e.g., [32]) explore the same change of perspective on images by trying to get the sentiment of filtered regions of the image to understand the bigger picture, but they also aggregate in the end. Finally, Abburi et al. [1] try to decompose music into parts (beginning and end) to understand the sentiment variance better. All these decomposed attempts had great results for their specific tasks, compared against their own baselines, even though in the end they aggregated the final product and only considered the singular values of the parts when measuring.

3. Information Theory and Entropy

The central concept of information theory, founded by Claude Shannon in 1948, is entropy, which measures the amount of uncertainty or randomness in a set of data. A higher entropy means there is more uncertainty, while a lower entropy means there is less. It provides the basis for lossless data compression, such as Huffman coding, and lossy data compression, such as JPEG and MP3 compression. These techniques are used to reduce the size of digital data without losing information.

Entropy is defined as the average amount of information required to describe an event or a set of data, with higher entropy corresponding to more uncertainty and lower entropy corresponding to less uncertainty [10]. For example, if we have a set of data that is completely predictable and has no uncertainty, the entropy of this data would be 0 bits. On the other hand, if the data is completely random and unpredictable, the entropy would be at its maximum.

Entropy is a concept from information theory that has applications in machine learning. In machine learning, entropy is used as a measure of impurity in a set of data. The entropy of a set of data is a measure of how mixed or homogeneous the data is [9]. If the entropy of the data is high, it will require more storage space and more bandwidth to transmit. On the other hand, if the entropy is low, the data can be stored and transmitted more efficiently.

In decision tree algorithms, such as ID3, C4.5, and CART, entropy is used to determine the best feature to split the data on at each step of the tree construction. The goal is to select a feature that will reduce the entropy of the data, making it more homogeneous and easier to classify [7].

The entropy is then used to determine the information gain, which is the reduction in entropy achieved by splitting the data based on that feature. The feature with the highest information gain is selected as the best feature to split the data on. The maximum entropy principle, first introduced by E.T. Jaynes in the 1950s, is a statistical principle that states that, given a set of constraints, **the probability distribution that best represents the underlying uncertainty is the one with the highest entropy**. In other words, the maximum entropy distribution is the one that has the most uncertainty or randomness, subject to the given constraints [24]. In machine learning, the maximum entropy principle is used in the training of maximum entropy models [25], such as logistic regression and maximum entropy Markov models. These models are trained to predict the probability of a set of outcomes based on a set of features, subject to the constraints imposed by the data. The goal is to find the distribution with the highest entropy that is consistent with the data.

4. Calculus Derivatives and Gradient

In calculus, the gradient is a vector that points in the direction of maximum increase of a given scalar-valued function. It can be thought of as a generalization of the concept of the derivative to higher dimensions.

Given a scalar-valued function $f(x)$ with $x = (x1, x2, ..., xn)$, the gradient of f at a point x is a vector of partial derivatives, denoted by $\nabla f(x)$ or $\nabla x f$. It is calculated as:

$\nabla f(x) = (df/dx1, df/dx2, ..., df/dxn)$

The gradient points in the direction of maximum increase of the function, so it is useful for finding the maximum or minimum values of the function. For example, if we want to find the maximum value of a function $f(x, y)$, we can find the gradient of the function at each point and move in the direction of the gradient until we reach a maximum [14].

The gradient is also used in gradient descent, an optimization algorithm commonly used in machine learning and other fields. In gradient descent, the parameters of a model are updated iteratively by taking a step in the direction of the negative gradient, which reduces the error in the model [21].

In vector calculus, the gradient of a scalar-valued function is used to define the concept of the divergence of a vector field, which measures the rate of change of a vector field in a given direction. The gradient is also used to define the concept of the curl of a vector field, which measures the amount of rotation in a vector field [11].

5. Figurative Languages and Sarcasm

Figurative language refers to the use of expressions or words in a non-literal sense, such as metaphors, similes, idioms, and sarcasm, in natural language [2]. Figurative language is a common feature of human communication and plays an important role in conveying emotions, opinions, and ideas.

In natural language processing (NLP), figurative language is a challenge for algorithms as it requires an understanding of context and the intention of the speaker. Unlike literal language, figurative language can have multiple interpretations and its meaning can change based on context [31].

To address this challenge, researchers in NLP have developed various techniques for detecting and interpreting figurative language in text. These techniques include:

- Metaphor detection: This involves identifying metaphorical expressions in text, such as "She has a heart of stone."
- Sarcasm detection: This involves detecting sarcastic expressions in text, such as "Great job, you really nailed it."
- Idiom recognition: This involves identifying idioms in text, such as Ït's raining cats and dogs."
- Simile detection: This involves identifying similes in text, such as "She sings like a bird."

These techniques can be applied to various NLP tasks, such as sentiment analysis, text classification, and question answering, to improve the accuracy of the results.

Sarcasm is a form of irony where the speaker's intended meaning is the opposite of what they actually say, normally by inverting the intention of the discourse [6]. Sarcasm detection is important for improving the accuracy of sentiment analysis and enhancing the overall understanding of natural language.

There are several approaches to sarcasm detection in NLP, including lexical, semantic, and contextual methods. Lexical methods rely on the use of specific words or phrases that are commonly associated with sarcasm, such as "Sure, that makes sense."Semantic methods focus on the meaning of the words and their relationships, while contextual methods use the context of the surrounding text to determine if a statement is sarcastic [31].

Humans communicate by humor as it delivers a serious message in popular and light way, therefore it is common to encounter such figurative language in social media, in services/product reviews as well as critiques to society and politics (even more on strict governments) [13].

So sarcasm is inherent present in our text and messages, even more in Brazil. Due to the strict control of media in past by military dictatorship and censuring the majority of artists had to resource to sarcasm in order to deliver subliminal messages of critiques in their works. This practice is ancient in Brazilian history and dates back from its colonial times. So it is only natural we would find a lot of sarcasm amongst Brazilian textual datasets [27].

6. Sentiment Gradient

Our proposed method changes the sentiment analysis representation. Instead of summarizing a message by a holistic sentiment (a number), we represent the message with its full nuances. The message is represented by a vector of sentences. Our proposal is inspired by the information theory of entropy, where we want to pass to the algorithms of machine learning the highest entropy representation of the text.

The traditional sentiment analysis representation works as the âBag-of-Featuresâ (BoF), which is the set of descriptors extracted from a textual message, denoted by A = bag of a_k, $k \in \{1, \ldots, N\}$, in which a_k is a feature and N is the total number of features in a message. For traditional sentiment analysis, each feature is a word that will have a sentiment score according to a function f, given by pre-defined mapping of words to a sentiment score. The average of those scores is the output of the Sentiment Analysis function (see Equation 1). Then, Sentiment(A) is one more feature considered to train the fake news classifier.

$$Sentiment(A) = \frac{1}{N}\sum_{k=1}^{N} f(a_k) \tag{1}$$

Instead of modeling the BoF of the message by their word components, we chose to model it by sentences because each sentence is an utterance about a target subject. Then, we propose a novel technique of applying derivatives into the array of features (see Equation 3), like what we would do on a time series (Equation 2), that way we would be able to capture the information we need, i.e., the rise, the fall and the stability of a sentimental gradient. Allowing classifiers to comprehend the sentiment variance of figurative languages such as Sarcasm. We named this new interpretation and technique Sentiment Gradient (Algorithm 1).

$$\mathrm{S(A)}{=}(Y_t : t \epsilon N) \tag{2}$$

$$Y_t = f(x) = \begin{cases} Y_1 = f(a_1), & i = 1 \\ Y_i = g(\frac{\partial f(a_{i-1})}{\partial sentiment}) & ,i > 1 \end{cases} \tag{3}$$

The sentences can be understood as the phases of our text signal, the sentiment gradient as the amplitude (which mathematically would indicate the intent of upward or downward change in sentiment), and the frequency is fixed by 20 sentences as it is the average in the dataset, as can be seen in Figure 1. To follow the trend, like the other works, we also have an aggregated average additional to our time-series. This measure is the average sentiment gradient of the series.

In the same figure, we can observe that the three classes of news fall in a specific spectrum of the signal, the True news maintain mostly the neutral to positive overtone, the fake news maintain mostly the negative impact overtone, and the sarcastic ones vary along the neutral point showing the subtle imbalance game to make us laugh.

Any given message that is not truncated by the communication channel is continuous on time, for word or sentence level. Therefore, by deriving a sentence's sentiment, we are not simply getting a singular value, but, depending on the mathematical signal, the direction to which the signal is flowing, because the derivative of a point 'a' shows us the tangent inclination of the variation of that point in relation to the linear function [4].

However, why choose to derive the sentences' time-series? Because the derivative function shows us the areas of increase, and decrease of a function as well as the magnitude of such change, passing more information than the singular value [12].

Algorithm 1: Sentiment Gradient Algorithm

```
Result: Sentiment Gradient of the News
sentiment_timeseries = empty array;
sentence_array = SentenceTokens(News);
if Length(sentence_array) ¿1 then
    for each sentence in sentence_array do
        sentiment_rate = sentence[sentiment_charge]\Length(sentence[tokens])
        sentiment_timeseries.append(sentiment_rate)
    end
    return mean(getGradients(sentiment_timeseries))
else
    return sentence_array[0][sentiment_charge]
end
```

With our approach we intend to increase the information entropy about the sentiment polarity phenomenon in text. Sarcasm is not a lie, but, instead a critic delivered in imbalanced polarity charged texts, meanwhile informative texts such as news deliver facts through monotonic less appealing writing. And fake news, lies, rely upon appealing language which exacerbates the tone (in any of the extremes).

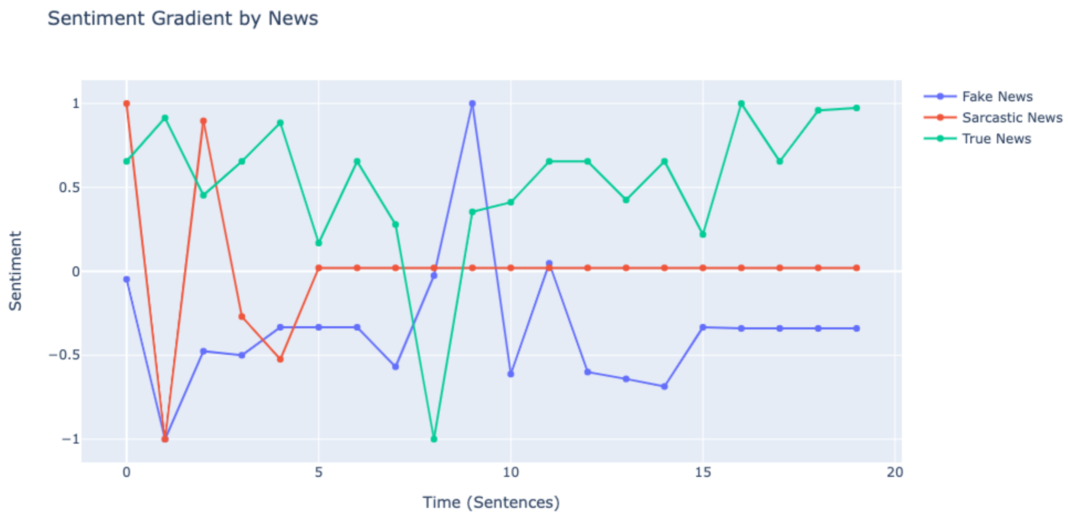

Figura 1: Example of sentiment gradient extracted from three different pieces of news with different tones: a fact (sentiment avg.: 0.5), a lie (sentiment avg.: -0.3), and a sarcastic (sentiment avg.: -0.1).

7. Dataset

We want the machine to understand the difference between ironic humorous critic, true facts and false statements. We expect the machine to perceive the nuances in sentiment charge across the sentences, and the style of writing (too dense text, too wordy, etc.).

Due to the lack of open datasets in Portuguese for both the fake and sarcastic news, we needed to leverage upon known popular sources of sarcastic and fake news in Brazil, E-Farsas' fake news session[1] and Sensacionalista[2]. Furthermore, for the True news, we chose Folha de SÃ£o Paulo[3], a journalistic source of news known to the Brazilian population. Although we have a full set of other sources for True News in Brazil, their sites have peculiarities and blockers that make the scraping process harder than Folha de SÃ£o Paulo. Also, we took advantage of a produced set of 2000 pair-wise true and fake news from the works of Silva et al. [30] and appended it into our main raw set to be preprocessed following our proposed methodology.

We considered news agencies and governmental accounts to outsource the truthful tweets. For the sarcastic ones, we chose popular sarcastic accounts, e.g., NÃ£o Salvo (@naosalvo), Sensacionalista (@sensacionalista) and O Criador (@OCriador). Finally, for fake news, we considered a set of reported known fake news spreader accounts, such as @opavao and current politics involved in the fake news scandal in Brazil from 2020 on-wards, which nonetheless, due to the extreme political propaganda material on its own, they can be classified as fake news as well as its unverified disseminated texts.

In order to extract the training set for our models, we created a web scraper for each of the sources. To do so, we relied upon Python programming language and its libraries, such as Beautifulsoup (lib for webscraping)[4], NLTK (natural language toolkit)[5] and re(for regular expression)[6]. Since our focus is on the textual content, we extracted only the textual news content from those sources and ignored extra media embedded into the news, e.g., videos, images and recordings.

We extracted the essential metrics from the textual set, like textual length and the average length of sentences. The procedure was similar and yet more straightforward for the tweets, as the API already

[1]E-Farsas - http://www.e-farsas.com/secoes/falso-2
[2]Sensacionalista - https://www.sensacionalista.com.br/
[3]Folha de SÃ£o Paulo - https://www.folha.uol.com.br/
[4]BeautifulSoup - https://www.crummy.com/software/BeautifulSoup/
[5]NLTK - https://www.nltk.org/
[6]Regular Expression - https://docs.python.org/3/library/re.html

returns the textual content, not needing the scraping part, only the preprocessing and feature extraction. In the end, our dataset has three classes, Fake, Sarcastic, and True News. With 76,782 rows of news prelabeled according to our strategy aforementioned.

8. Experiment

The most used features by the literature for the sentiment analysis [5] are Word Sentimental Score and Sentence Sentimental Score. However, different from our novelty, this step is usually focused only on the singular synthesized metric of sentiments (sum or average) from the entire text or sentence. For the basic features the literature recommend word count, sentence count, space count, and POS Tag count.

We investigated the difference between the basic sentiment analysis and the Sentiment Gradient combined with basic features (results in Table 8). In order to fairly compare against the related works, we trained models with all the features, analogously they did in their respective works (results in Table 9). The only public dataset in Portuguese we found available was the Fake BR, that we used to train the same baseline SVM model of [19], which we obtained the same 0.89 % score to compare against our own.

We ran the experiment of cross validating (using 5 folds, repeated 10 times) each one with each set of the features we engineered as well as combinations of the basic and each approach for sentiment analysis. After this process we obtained distributions of 10 repetitions of 5, i.e., 50 registries to which we compared against each other through Mann Whitney U Hypothesis test to check for statistical significance in our results (obtaining p-values lower than 0.05 for all comparisons).

8.1. Machine Learning Algorithms and Hyperparameters

The state of the art in fake news Detection indicates the following techniques as most used ones [29]: K-Nearest Neighbors (KNN), Naive Bayes (Gaussian, GNB, and Multi-nominal, MNB), Decision Tree, Random Forest (R.For.), Adaboost Tree, Gradient Boosting, Support Vector Machine (SVM), Linear Regression Classifier (LNR), LSTM and Multi-Layer Perceptron (MLP).

The following algorithms used the hyperparameter tuning process of 5 fold cross validation, using f1-weighted as the comparison metric to decide the best hyperparameter configuration. This tuning process used the existing Gridsearch of Python's SKLearn Library.

- **KNN**:number of neighbors as 5, weighting strategy of euclidean distance, and deciding algorithm of ball tree.
- **Decision tree**:max depth of 10, and both minimum leaf samples, and minimum split samples of 2. With a random state of zero, for replication purposes.
- **Random Forest**:500 estimators in the forest, each base classifier with max depth of 40, min samples in the leaf nodes of 1, and min samples of split of 2. With a random state of zero, for replication purposes.
- **Gradient Boosting**:500 base estimators, 0.01 learning rate, max depth of 10, min samples in leaf of 1, and min samples of split as 2. For the Adaboost classifier, we got as best configuration set: 500 base estimators, and learning rate of 0.01.
- **Multi-Layer Perceptron**:Adam as solver, alpha of 1-e5, constant learning rate, activation function of relu, and the following architecture tuple (10,20,30,40,50,40,30,20,10).
- **LSTM**:activation function of hyperbolic tangent, optimizer adam, loss of sparse categorical crossentropy, and the following architecture tuple (12 Dense, 4 Recurrent Units, Flatten, and Softmax, each hidden layer with a subsequent dropout of 20 %).

For the **Linear Regression** algorithm, **Support Vector Machine**, **Gaussian Naive Bayes**, and **Multinomial Naive Bayes** we configure as studied from literature because of the simplicity of naive bayes algorithms and SVM and Linear Regression algorithms' complexity when using the probability functions (causing computer resource overload).

Model	Approach	Score
GNB	Basic + Sentiment	0.864 (+/- 0.004)
GNB	Basic + SentimentGradient	0.715 (+/-0.004)
LSTM	Basic + Sentiment	0.887 (+/-0.004)
LSTM	Basic + SentimentGradient	0.890 (+/-0)
R.For.	Basic + Sentiment	0.890 (+/-0)
R.For.	Basic + SentimentGradient	0.890 (+/-0)
SVM	Basic + Sentiment	0.890 (+/-0)
SVM	Basic + SentimentGradient	0.890 (+/-0)

Cuadro 1: Twitter Dataset Cross-Validation Metrics.

Work	Winning Model	Score
Our Approach with News	Grad. Boosting Classifier	0.949
Our Approach with Tweets	R.For.	0.893
Wang et al. 2018	Logistic Regression	0.942
Bhutani et al. 2019	CNN	0.847
Manjushaa and Raseek, 2018	CNN	0.797
Monteiro et al., 2018	SVM	0.890
de Morais et al., 2019	LP	0.800

Cuadro 2: Comparison between our approach and the related works, compared against the Tweets classification score.

- **Linear Regression Classifier** with multi-class parameter set to multinomial
- **SVM** with probability option set to True, and divided in 6 jobs
- **Gaussian Naive Bayes** with standard parameters
- **Multinomial Naive Bayes** with standard parameters

8.2. Study Case in Twitter

We wanted to experiment how our approach would behave on messages coming from Twitter. In order to do this, what we did was to repeat the data gathering part of our experiment, now this time for Twitter.

In order to preemptively obtain the label values of each tweet, we took profiles known for spreading messages of each category (Fake, Sarcastic and True messages) and composed our dataset from there. We ran the exact same preprocessing we ran for the news set on the twitter-set, obtaining the same features.

The key point of analysis here is that the sentiment gradient on tweets would operate in a short number of sentences, due to Twitter's post size restriction (240 characters only). The average number of sentences in a tweet is 2. And following our algorithm the remaining pads (18 right pads) would be attenuated with 0. This is a limitation of the classical models, in the sense that all features should be fixed, linear, and normalized.

Then, we repeated the cross-validation experiment with the same parameters, 5 folds, repeating 10 times. In Table 1 we can see better the metrics we got.

From the results in this subsection we can see that more studies and experimentation is needed to make sentiment gradient as effective to tweets as it was for news.

Also for the joint of all attributes, we can see it's metrics in Table 2

9. Experiment Results Discussion

As shown in Table 8, the Sentiment Gradient feature positively impacted most of the results of the model trained. We can observe an increase of 14 % for Random Forests, and 6 % for Gradient Boost Classifiers for the best models.

Model	**Feature Choice**	F1(+/-Stdv)
Adaboost	Basic + Sentiment	0.736(+/-0.007)
Adaboost	Basic + SentimentGradient	0.739(+/-0.007)
DecTree	Basic + Sentiment	0.757(+/-0.007)
DecTree	Basic + SentimentGradient	0.754(+/-0.008)
GNB	Basic + Sentiment	0.612(+/-0.019)
GNB	Basic + SentimentGradient	0.594(+/-0.011)
GradientBoost	Basic + Sentiment	0.778(+/-0.005)
GradientBoost	**Basic + SentimentGradient**	**0.832(+/-0.008)**
KNN	Basic + Sentiment	0.748(+/-0.007)
KNN	Basic + SentimentGradient	0.661(+/-0.008)
LNR	Basic + Sentiment	0.551(+/-0.003)
LNR	Basic + SentimentGradient	0.632(+/-0.007)
LSTM	Basic + Sentiment	0.656(+/-0.016)
LSTM	Basic + SentimentGradient	0.677(+/-0.011)
MLP_ADAM	Basic + Sentiment	0.756(+/-0.013)
MLP_ADAM	Basic + SentimentGradient	0.769(+/-0.012)
MNB	Basic + Sentiment	0.24(+/-0.000)
MNB	Basic + SentimentGradient	0.24(+/-0.000)
R.For.	Basic + Sentiment	0.788(+/-0.007)
R.For.	**Basic + SentimentGradient**	**0.846(+/-0.006)**
SVM	Basic + Sentiment	0.554(+/-0.005)
SVM	Basic + SentimentGradient	0.577(+/-0.008)

Cuadro 3: F1 Measure comparing the usage of regular sentiment analysis and sentiment gradient. As seen in bold, the best performing models were the Random Forest and Gradient Boosting Classifiers setup with Sentiment Gradient.

We experimented our model on tweets, wondering whether the data type would influence the results. Employing our scrapers, we were able to outsource tweets about the same topics of our news from known reference profiles on the classes (Fake, True, and Sarcastic), obtaining 80,843 tweets. We observed that due to the lack of sentences in Tweet's microblog structure the sentiment gradient is unable to get the nuance because the granularity would need to change to world level maybe, as our results for that kind of data were of 89 % F-Measure.

Even though the MLP and LSTM are modern approaches from the neural networks class of algorithms, they were not able to beat the ensemble methods, and this may be explained by the lack of data that we have nowadays in Portuguese just yet. However, analyzing on the contribution of the sentiment gradient, those models were impacted positively as well.

On the other hand, we see that GNB, Decision Tree, and KNN models got inverted results, benefiting more from the traditional sentiment analysis than the sentiment gradient. We attribute this behavior to the increase in the data dimensionality as we consider twenty sentences for representing the sentiment gradient, but KNN and Decision Tree rely on their linear separation of observations' space, a strategy that can be harmed by higher dimension sets.

We ran the cross-validation training to perceive the real impact of sentiment gradient. We defined a 10 folded cross-validation setup on F1 Measure and applied the Mann Whitney U Test to check if the results from the cross val distributions were statistically significant. We got P-Values inferior to the threshold of 0.05, i.e., our results have a confidence interval of 95 %.

Comparing our results with the related works (see Table 9), we understand that our approach strongly contributes to the Brazilian Portuguese differentiation of True, Fake, and Sarcastic News, as we achieved more than 5 % in relation to similar works. Although we are not in the same language context, we surpassed the two CNN works and got pretty close to the best English detection work with a difference of only 0.7 %. We attribute this success to the fact that our classifiers were able to comprehend the nuances of sentences' sentiment charge different from the related works handling a singular fixed value (concentrated by an average, rather than the expansion of the series), and also due to the information

Work	Winning Model	Score
Our Approach	**Gradient Boosting Classifier**	**0.949**
Wang et al. 2018	Logistic Regression	0.942
Bhutani et al. 2019	CNN	0.847
Manjusha and Raseek, 2018	CNN	0.797
Monteiro et al. 2018	SVM	0.89
de Morais et al, 2019	LP	0.80

Cuadro 4: Comparison between our approach and the related works.

that the derivative function brings us, i.e., the direction of change and its magnitude.

Comparing the results for the application of our approach over tweets, we can see that the solution was not as effective since we could see that for some algorithms such as the Random Forest and SVM the inclusion of it seems not to make difference, and for GNB and LSTM the traditional sentiment analysis worked better than sentiment gradient. Also in Table 2 we can see that even though we got competitive metrics against the related works, it is not as effective as the joint set for news, implying that the sentiment gradient didn't impact the classification of tweets.

We credit this for the fact that tweets have a much different structure if compared to news, they are smaller, and with more slang and abbreviations. For the sentiment gradient to work with tweets, we may need to change the algorithm for taking words in consideration rather than sentences, however, this can be challenging as well, since there tweets that don't rely much on writing and instead on links, or medias.

Our results show a benefit of using Sentiment Gradient on the fake news detection process. However, more experimentation on other language datasets is recommended, as well as exploring other format of news such as microblogs that might require mathematical modeling experimentation in the sentiment gradient approach (from sentences to words) in order to get results closer to the ones in news analysis.

10. Experimenting the Sentiment Gradient Sizing

Algorithm 1 deals with four scenarios to best fit the sentence size of the text in the dataset in a uniform representation, so that every registry has the same size. In the first scenario, the length of the sentence is greater than one and equal to the sentiment gradient sizing (sent_grad_sizing, or SGS) - then we calculate the gradients for each sentence and return the new timeseries representation. In the second scenario, the number of sentences of the given text is fewer than the SGS, so we need to pad the sentence to be consistent with the dataset. In this case we put the neutral value of 0 for the remaining pads. When the number of sentences is greater than the SGS, we complete timeseries calculation and resample it to fit within SGS, by calculating the ratio between the number of sentences and the SGS and resplicing the timeseries by it. Then, we apply mean statistics through the splits of the new splice (e.g., should the SGS be equal to four and the number of sentences 16, and we would have a new timeseries of four sentences in each; it is a mean of the gradient of sections of four sentences of the original text), thus compacting. If the SGS equals one, it is the maximum compacting, collapsing the entire original timeseries into an average gradient. Finally, if the text has only one sentence, it returns its singular mean sentiment score.

One question still lingers, how does the sentiment gradient sentence sizing influence the results? In order to answer that, we execute an experiment to check the benefit of using this new sentiment analysis representation and discover how much affected it would be by the data.

10.1. Sentiment Gradient Sizing Experiment Setup

In this study, we wanted to define the best sizing of the gradient scale. In order to do this, we varied the algorithm of sentiment series construction from 1 (maximum compacting) to 50 (max number of sentences in the set). Also, we marked as -1 the traditional sentiment analysis.

That given, we constructed a function of repeated k fold experiment sequence to test which would be the best sequencing size. The repetition is executed 10 times each for a kfold of 5 for each sequencing size then compared to know which sizing gave the best metric output for a set of algorithms used in literature. The used metric to evaluate the ideal sizing was the F1-Measure Micro.

Sent. Sampling: [1,50]			Sent. Sampling: [1,25]			Sent. Sampling: [26,50]		
Model	SGS	F1 Score	Model	SGS	F1 Score	Model	SGS	F1 Score
Dec.Tree	-1	0.453(+/-0.073)	Dec.Tree	-1	0.478(+/- 0.084)	Dec.Tree	-1	0.522(+/-0.016)
Dec.Tree	1	**0506(+/-0)**	Dec.Tree	1	**0.510(+/- 0)**	Dec.Tree	1	0.500(+/-0)
Dec.Tree	16	0.475(+/-0.136)	Dec.Tree	13	0.503(+/- 0.143)	Dec.Tree	25	**0.524(+/-0.024)**
GNB	-1	0.478(+/-0.056)	GNB	-1	0.510(+/- 0)	GNB	-1	0.523(+/-0.015)
GNB	1	**0.506(+/-0)**	GNB	1	**0.510(+/- 0)**	GNB	1	0.500(+/-0)
GNB	2	0.506(+/-0)	GNB	3	0.497(+/- 0.053)	GNB	21	**0.525(+/-0.029)**
KNN	-1	0.399(+/-0.046)	KNN	-1	0.364(+/- 0.049)	KNN	-1	0.511(+/-0.009)
KNN	1	0.424(+/-0.165)	KNN	1	0.435(+/- 0.149)	KNN	1	0.402(+/-0.197)
KNN	25	**0.499(+/-0.082)**	KNN	50	**0.496(+/- 0.112)**	KNN	18	**0.516(+/-0.007)**
LNR	-1	0.500(+/-0.023)	LNR	-1	0.501(+/- 0.017)	LNR	-1	0.479(+/-0.020)
LNR	1	**0.506(+/-0)**	LNR	1	0.501(+/- 0)	LNR	1	0.500(+/-0)
LNR	14	0.498(+/-0.082)	LNR	12	**0.517(+/- 0.104)**	LNR	31	**0.522(+/-0.016)**
R.For.	-1	0.447(+/-0.079)	R.For.	-1	0.486(+/-0.076)	R.For.	-1	0.525(+/-0.009)
R.For.	1	**0.506(+/-0)**	R.For.	1	**0.510(+/-0)**	R.For.	1	0.500(+/-0)
R.For.	2	0.459(+/-0.134)	R.For.	18	0.508(+/-0.156)	R.For.	17	**0.556(+/-0.025)**

Cuadro 5: F1 Score to detect the ideal sentiment gradient size (SGS).

Sent. Sampling:[1,50]						Sent. Sampling:[1,25]						Sent. Sampling:[26,50]					
Model	SGS	F1	Precision	Recall	Accr	Model	SGS	F1	Precision	Recall	Accr	Model	SGS	F1	Precision	Recall	Accr
R.For.	-1	0447	0.516	0.516	0.516	R.For.	-1	0.486	0.527	0.527	0.527	R.For.	-1	0.520	0.530	0.530	0.530
R.For.	1	**0.506**	0.503	0.503	0.503	R.For.	1	**0.510**	0.508	0.508	0.508	R.For.	1	0.500	0.500	0.500	0.500
R.For.	2	0.459	**0.529**	**0.529**	**0.529**	R.For.	18	0.508	**0.568**	**0.568**	**0.568**	R.For.	17	**0.556**	**0.564**	**0.564**	**0.564**

Cuadro 6: Auxiliary metrics of the Best-Performing Algorithm in Sizing Experiment

The chosen algorithms for this experiment were the KNN, Logistic Regression, Gaussian Naive Bayes, Decision Tree, and Random Forest. Each algorithm was configured with its best parameters through hyperparameter tuning tests, given the following configurations for each:

- Dec.Tree - max_depth = 15, criterion=Entropy
- GNB - var_smoothing = 0.8
- KNN - k = 5
- LNR - solver = sag, penalty = l2
- R.For. - max_depth = 10, n_estimators = 300, criterion = Gini

First, we ran the experiment for the entire dataset. However, we noticed that the model's metrics were affected by the number of sentence paddings chosen. So, we decided to organize our experiment in three scenarios: all (i.e., for all registries), less than 25 (i.e., a dataset with registries that had less than 25 sentences per text), and more than 25 and less than 50 (i.e., registries whose text had sentences in between that interval).

After this experiment, we have obtained the results shown in Table 5. There is a benefit in using the approach rather than the traditional sentiment analysis, but the method is sensitive to the sentiment gradient sizing, i.e., we obtain different values for max compacting (1) or other expansions of the formula.

We also analyzed the precision, recall, and accuracy of the best-performing algorithm (Table 6).

10.2. Sentiment Gradient Sizing Experiment Results

10.2.1. Padding for All Samples

The padding experiment for all samples consists in training the chosen algorithms with the entire dataset and observing which sentiment gradient sizing is the best. The Gaussian Naive Bayes algorithm performed better for this sampling, whereas the other algorithms suffered from the fact that we did not stratify the dataset by the number of sentences.

The algorithm that required the longest expansion of the sentiment gradient was the KNN with 25 paddings. On the other hand, the lowest was the GNB and Random Forest with only 2. It can be explained by the fact that R.For. and GNB penalize too many features (even more with a strong correlation between them).

In general, what we can observe from that sampling is that the maximum compaction of the sentiment gradient was more effective than its expansion, which is related to the stratification, as the maximum compacting transforms the output of the sentiment gradient into a single value of the direction of the sentiment charge of the text.

10.2.2. Padding for Less than 25 Sentences Samples

The padding experiment for all registries with less than 25 sentences attempts to understand what could happen if we simply stratify the dataset and retrain the algorithms. We expect to see two things: the improvement of metrics and leverage on the wider expansion of the sentiment gradient.

As we can observe, the algorithms leveraged more on an expanded sentiment gradient than the prior extract. However, some cases (Dec.Tree, GNB, and R. For.) obtained better results with a maximum compacting.

There was an improvement in the metrics compared to the prior extract, implying that by stratifying the training step, we are going in the right direction. The logistic regression and Random Forest show it clearly.

10.2.3. Padding for Samples with more than 25 Sentences and Less than 50

The padding experiment for more than 25 sentences was the second stratification part of our check, and as much as the extract from 1 to 25, we expect some metric improvement and more usage of the expanded sentiment gradient.

As far as we can see in Table 5, the SGS that was not equal to maximum compacting (1) has used much more of the expanded sentiment gradient than the prior extracts, and all expanded results were better than maximum compacting and better than traditional sentiment analysis.

The highest score goes to the Random Forest Algorithm with an expanded gradient of 17 paddings, obtaining an average of 55 % of F1-Measure, which is very interesting if we highlight that the algorithm was trained with only one feature set (i.e., traditional sentiment analysis and the class, or sentiment gradient paddings and the class).

11. Proportion Experiment

The limitation of our algorithm implementation is the sentence size limit to calculate the gradient. Then, we've run the proportion experiment, where we created a dataset for each variation of limit within the range of the number of sentences available in the original corpus.

Since we have a distribution of 1 to 28 sentences, for each sentence size set we got different dataframes with varying sentiment gradient for each varying from 1 (maximum collapsing) till the max sentence size possible (maximum expansion), also we had for each the traditional sentiment analysis to compare against. Totaling 252 sets, upon which we ran a random forest algorithm to try to classify the registries into True, Sarcastic or Fake News.

The Random Forest had its hyperparameters defined thorough GridSearchCV process resulting in the following setting: max_depth of 5, 500 estimators, Gini criterion, min samples split of 2 , min samples leaf of 3, min weight fraction of 0.05, max features defined by square root function, random seed set to 0, and number of jobs to -1 (i.e., running on all available computing resources).

This experiment consists of using only sentiment gradient features (the steps of the timeseries) with the class column for training the classifier algorithm, then we extract the weighted f-measure of the cross validated 5 fold run. We do the same for the traditional sentiment analysis singular feature. Then we rank only the best results for the sentiment gradient for each maximum number of sentences, to map which was the best proportion of sentiment gradient sizing. After that we calculate the difference in F1 between the novelty approach and traditional one to determine the gain of using the sentiment gradient.

sentences qty	sent. grad. size	F1	proportion	F1 gain
1	1	0.79	100 %	-3.1 %
2	2	0.73	100 %	-5.5 %
3	3	0.78	100 %	+3 %
4	2	0.76	50 %	+3.7 %
5	2	0.77	40 %	+7 %
6	6	0.77	100 %	+6.6 %
7	7	0.77	100 %	+7.3 %
8	8	0.77	100 %	+8 %
9	9	0.77	100 %	+8.4 %
10	10	0.76	100 %	+8.8 %
11	11	0.76	100 %	+8.2 %
12	12	0.75	100 %	+8.5 %
13	12	0.75	92 %	+8.2 %
14	4	0.75	29 %	+9 %
15	4	0.74	27 %	+10.3 %
16	14	0.74	88 %	+12 %
17	17	0.73	100 %	+13.3 %
18	17	0.72	94 %	+13 %
19	19	0.71	100 %	+12.8 %
20	20	0.70	100 %	+12.1 %
21	10	0.69	48 %	+11.1 %

Cuadro 7: Proportion Experiment Results ran with Random Forest. Columns: maximum number of sentences in the subset, size of the sentiment gradient expansion, F1 weighted metric monitored, the proportion of the sentiment gradient sizing with respect to the maximum number of sentences in the subset, and finally the gain of using the sentiment gradient in relation to the traditional analysis measured in F1 weighted and expressed as positive or negative gain.

After that we collected the f-measure we observed (Table 7) that first, there was indeed gain in using the gradient over traditional sentiment analysis, which entails with our main theory. But, with this experiment we were able to understand which would be the ideal proportion of sentence size and gradient limit size. The most frequent proportion was of 100 %, i.e. total expansion of the sentiment, peaking with a gain of 13.3 % in F1 (for 17 sentences in text). The average proportion according to our table of best results would be of 84.19 % of maximum number of sentences in the dataset.

12. Discussion

As shown in Table 8, the Sentiment Gradient feature positively impacted most of the results of the model trained. We can observe an increase of 14 % for Random Forests, and 6 % for Gradient Boost Classifiers for the best models.

We experimented our model on tweets, wondering whether the data type would influence the results. Employing our scrapers, we were able to outsource tweets about the same topics of our news from known reference profiles on the classes (Fake, True, and Sarcastic), obtaining 80,843 tweets. We observed that due to the lack of sentences in Tweet's microblog structure the sentiment gradient is unable to get the nuance because the granularity would need to change to world level maybe, as our results for that kind of data were of 89 % F-Measure.

Even though the MLP and LSTM are modern approaches from the neural networks class of algorithms, they were not able to beat the ensemble methods, and this may be explained by the lack of data that we have nowadays in Portuguese just yet. However, analyzing on the contribution of the sentiment gradient, those models were impacted positively as well.

On the other hand, we see that GNB, Decision Tree, and KNN models got inverted results, benefiting more from the traditional sentiment analysis than the sentiment gradient. We attribute this behavior to the increase in the data dimensionality as we consider twenty sentences for representing the sentiment gradient, but KNN and Decision Tree rely on their linear separation of observations' space, a strategy that can be harmed by higher dimension sets.

We ran the cross-validation training to perceive the real impact of sentiment gradient. We defined

Model	Feature Choice	F1
Adaboost	Basic + Sentiment	0.736(+/-0.007)
Adaboost	Basic + SentimentGradient	0.739(+/-0.007)
DecTree	Basic + Sentiment	0.757(+/-0.007)
DecTree	Basic + SentimentGradient	0.754(+/-0.008)
GNB	Basic + Sentiment	0.612(+/-0.019)
GNB	Basic + SentimentGradient	0.594(+/-0.011)
GradientBoost	Basic + Sentiment	0.778(+/-0.005)
GradientBoost	**Basic + SentimentGradient**	**0.832(+/-0.008)**
KNN	Basic + Sentiment	0.748(+/-0.007)
KNN	Basic + SentimentGradient	0.661(+/-0.008)
LNR	Basic + Sentiment	0.551(+/-0.003)
LNR	Basic + SentimentGradient	0.632(+/-0.007)
LSTM	Basic + Sentiment	0.656(+/-0.016)
LSTM	Basic + SentimentGradient	0.677(+/-0.011)
MLP_ADAM	Basic + Sentiment	0.756(+/-0.013)
MLP_ADAM	Basic + SentimentGradient	0.769(+/-0.012)
MNB	Basic + Sentiment	0.24(+/-0.0)
MNB	Basic + SentimentGradient	0.24(+/-0.0)
R.For.	Basic + Sentiment	0.788(+/-0.007)
R.For.	**Basic + SentimentGradient**	**0.846(+/-0.006)**
SVM	Basic + Sentiment	0.554(+/-0.005)
SVM	Basic + SentimentGradient	0.577(+/-0.008)

Cuadro 8: F1 Metrics comparing the usage of regular sentiment analysis and sentiment gradient. As seen in bold, the best performing models were the Random Forest and Gradient Boosting Classifiers setup with Sentiment Gradient.

Work	Winning Model	Score
Our Approach	**Gradient Boosting Classifier**	**0.949**
Wang et al. 2018	Logistic Regression	0.942
Bhutani et al. 2019	CNN	0.847
Manjusha and Raseek, 2018	CNN	0.797
Monteiro et al. 2018	SVM	0.89
de Morais et al, 2019	LP	0.80

Cuadro 9: Comparison between our approach and the related works.

a 10 k folded cross-validation setup on F1 Measure and applied the Mann Whitney U Test to check if the results from the cross val distributions were statistically significant. We got P-Values inferior to the threshold of 0.05, i.e., our results have a confidence interval of 95 %.

Comparing our results with the related works (see Table 9), we understand that our approach strongly contributes to the Brazilian Portuguese differentiation of True, Fake, and Sarcastic News, as we achieved more than 5 % in relation to similar works. Although we are not in the same language context, we surpassed the two CNN works and got pretty close to the best English detection work with a difference of only 0.7 %. We attribute this success to the fact that our classifiers were able to comprehend the nuances of sentences' sentiment charge different from the related works handling a singular fixed value (concentrated by an average, rather than the expansion of the series), and also due to the information that the derivative function brings us, i.e., the direction of change and its magnitude.

Comparing the results for the application of our approach over tweets, we can see that the solution was not as effective since we could see that for some algorithms such as the Random Forest and SVM the inclusion of it seems not to make difference, and for GNB and LSTM the traditional sentiment analysis worked better than sentiment gradient. Also in Table 2 we can see that even though we got competitive metrics against the related works, it is not as effective as the joint set for news, implying that the sentiment gradient didn't impact the classification of tweets.

We credit this for the fact that tweets have a much different structure if compared to news, they are smaller, and with more slang and abbreviations. For the sentiment gradient to work with tweets, we may need to change the algorithm for taking words in consideration rather than sentences, however, this can be challenging as well, since there tweets that don't rely much on writing and instead on links, or medias.

Our results show a benefit of using Sentiment Gradient on the fake news detection process. However, more experimentation on other language datasets is recommended, as well as exploring other format of news such as microblogs that might require mathematical modeling experimentation in the sentiment gradient approach (from sentences to words) in order to get results closer to the ones in news analysis.

The proportion experiment served to show us that the sentence size indeed impacts the method. From our observations (Table 7) we could understand that the best approach most of the times would be to use the full expansion of 100 %, peaking with 13.3 % of gain in relation to the traditional approach. Although for the greatest number of sentences in our corpus being 21 sufficed with 48 % of proportion with 11.1 % of gain.

12.1. Findings & limitations

We presented the novelty of unpacking the traditional sentiment analysis feature that is approached by other studies as a singular attribute only, to time-series formatting. We attribute the success of this proposal to the principle of Max Entropy where the distribution that best represents a phenomenon is the one with more entropy in it.

The sentiment gradient, through the vector of derivatives of the sentences' polarities of the text, is capable of mapping the sentiment polarity charge imbalance inherent of the humorous text writing in sarcasm figurative language.

On top of that, we saw in literature that figurative languages in general, and mainly sarcasm are the preferred way of delivering review about aspects of politics and society. In more restrictive governments this is the only option even.

Since we are working with a Brazilian Portuguese Corpus, the sarcasm differentiation is very useful, due to the cultural and historical presence of sarcasm in Brazilian textual works, and news about politics and society. Not only that, it is known from literature in Fake News detection that True/Factual news are less appealing in polarity and more monotonic, and Fake News are more appealing to the extremes of the sentiment polarity scoring, so the entropy increase helps us on understanding each document as a vector of derivatives, or a timeseries.

We have the limitation of the sizing of the gradient in order to comply with current machine learning algorithms (to have standard features, i.e. same number of features, for all training examples). However, this is also contemplated by the information theory, whereas we want to know the best way to compress information to not loose relevant information. So we compress documents with number of sentences greater than the limit by applying the same sentiment gradient algorithm to the splices of the document. And we tackled the proportion sentiment gradient sizing to better choose the limit, much similar to what is done to elbow method in K-Means clustering.

Although the language is a limitation in the sense that each culture and language have unique idioms, ways of representing sarcasm, we know that the foundation of sarcasm is the inversion of sentiment polarity in between tokens of the document (in this study, sentence tokens). The fact that sarcasm is rooted in Brazilian writing due to the historical and cultural importance of it to our ideological manifestations, makes our dataset a very adequate studying ground to research about the impact of differentiating sarcastic texts from facts and lies, as well as helps us on evaluating different approaches to sarcasm detection.

Finally we attribute the increase in monitoring metrics from Sentiment Gradient Only, Basic plus Sentiment Gradient, and JudiceVerum in relation to the Traditional Sentiment Analysis, Basic Plus Traditional Sentiment, and other works described in literature is the fact that each time we increase entropy more. Culminating in the usage of all features (traditionally explored in literature already, such as POS-Tagging) with our novelty (Sentiment Gradient). Which makes a lot of sense since one of the most used techniques in literature for sarcasm detection is to map the POS-Tagging frequency to the label of sarcasm, therefore increasing entropy on top of this improved the complete feature set, as intended by the principle of maximum entropy.

13. Conclusions

From the results we got in our experiments, we were able to test our proposed novelty of the sentiment gradient being effective to help differentiate what is fake, true, or sarcastic as it can provide to the machine the sentimental imbalance that occurs in sarcastic cues.

The current limitation of the novelty is the fact that it is not as effective for tweets due to their different writing style, which relies on faster, smaller, and more direct sarcastic cues. Furthermore, more experiments on other language datasets are suggested, as we restricted our experiments to the Brazilian Portuguese Scenario.

We see as contributions of this work the sentiment gradient technique we propose and the dataset which allowed us to do such study, as Portuguese sets of labeled news are hard to find due to the commodity of using known English sets.

For future works, we intend to expand the sentiment gradient concept to other datasets, such as the tweets, and also explore other machine learning algorithms besides the classic models.

Referencias

[1] Abburi, H., Akkireddy, E.S.A., Gangashetti, S., Mamidi, R.: Multimodal sentiment analysis of telugu songs. In: SAAIP@ IJCAI (2016)

[2] Allo, M.D.G.: Figurative languages by the lecturers in teaching english at english study program of fkip uki toraja. TEFL Overseas Journal Christian University Of Indonesia Toraja **1**(1), 76–89 (2015)

[3] Bhutani, B., Rastogi, N., Sehgal, P., Purwar, A.: Fake news detection using sentiment analysis. In: 2019 Twelfth International Conference on Contemporary Computing (IC3). pp. 1–5. IEEE (2019)

[4] Bos, H.J.M.: Differentials, higher-order differentials and the derivative in the leibnizian calculus. Archive for history of exact sciences **14**(1), 1–90 (1974)

[5] Charalampakis, B., Spathis, D., Kouslis, E., Kermanidis, K.: A comparison between semi-supervised and supervised text mining techniques on detecting irony in greek political tweets. Engineering Applications of Artificial Intelligence **51**, 50–57 (2016)

[6] Cheang, H.S., Pell, M.D.: The sound of sarcasm. Speech communication **50**(5), 366–381 (2008)

[7] Coifman, R.R., Wickerhauser, M.V.: Entropy-based algorithms for best basis selection. IEEE Transactions on information theory **38**(2), 713–718 (1992)

[8] CORDEIRO, P.R.D., Pinheiro, V., Moreira, R., Carvalho, C., Freire, L.: What is real or fake?-machine learning approaches for rumor verification using stance classification. In: IEEE/WIC/ACM International Conference on Web Intelligence. pp. 429–432. ACM (2019)

[9] Du, M., Wang, S.M., Gong, G.: Research on decision tree algorithm based on information entropy. In: Advanced Materials Research. vol. 267, pp. 732–737. Trans Tech Publ (2011)

[10] Fulcher, B.D.: Feature-based time-series analysis. In: Feature engineering for machine learning and data analytics, pp. 87–116. CRC Press (2018)

[11] Garg, V., Singh, K.: An improved grunwald-letnikov fractional differential mask for image texture enhancement. International journal of advanced computer science and applications **3**(3) (2012)

[12] Górecki, T., Łuczak, M.: Using derivatives in time series classification. Data Mining and Knowledge Discovery **26**(2), 310–331 (2013)

[13] King, G., Pan, J., Roberts, M.E.: How censorship in china allows government criticism but silences collective expression. American political science Review **107**(2), 326–343 (2013)

[14] Li, S., Meng, K., Penot, J.P.: Calculus rules for derivatives of multimaps. Set-Valued and Variational Analysis **17**(1), 21–39 (2009)

[15] Luo, L., Ao, X., Pan, F., Wang, J., Zhao, T., Yu, N., He, Q.: Beyond polarity: Interpretable financial sentiment analysis with hierarchical query-driven attention. In: IJCAI. pp. 4244–4250 (2018)

[16] Manjusha, P., Raseek, C.: Convolutional neural network based simile classification system. In: 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR). pp. 1–5. IEEE (2018)

[17] Manzoor, S.I., Singla, J., et al.: Fake news detection using machine learning approaches: A systematic review. In: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). pp. 230–234. IEEE (2019)

[18] Marchi, R.: With facebook, blogs, and fake news, teens reject journalistic âobjectivityâ. J. Commun. Inq. **36**(3), 246–262 (2012)

[19] Monteiro, R.A., Santos, R.L., Pardo, T.A., de Almeida, T.A., Ruiz, E.E., Vale, O.A.: Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In: International Conference on Computational Processing of the Portuguese Language. pp. 324–334. Springer (2018)

[20] de Morais, J.I., Abonizio, H.Q., Tavares, G.M., da Fonseca, A.A., Barbon Jr, S.: Deciding among fake, satirical, objective and legitimate news: A multi-label classification system. In: Proceedings of the XV Brazilian Symposium on Information Systems. p. 22. ACM (2019)

[21] Pflug, G.C., Weisshaupt, H.: Probability gradient estimation by set-valued calculus and applications in network design. SIAM Journal on Optimization **15**(3), 898–914 (2005)

[22] Pinto, M.R., de Lima, Y.O., Barbosa, C.E., de Souza, J.M.: Towards fact-checking through crowdsourcing. In: 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD). pp. 494–499. IEEE (2019)

[23] Polage, D.C.: Making up history: False memories of fake news stories. Eur. J. Psychol. **8**(2), 245–250 (2012)

[24] Rajagopalan, V., Ray, A., Samsi, R., Mayer, J.: Pattern identification in dynamical systems via symbolic time series analysis. Pattern Recognition **40**(11), 2897–2907 (2007)

[25] Ratnaparkhi, A.: A simple introduction to maximum entropy models for natural language processing. IRCS Technical Reports Series p. 81 (1997)

[26] Reis, J., Correia, A., Murai, F., Veloso, A., Benevenuto, F.: Explainable machine learning for fake news detection. In: Proceedings of the 10th ACM Conference on Web Science. pp. 17–26. ACM (2019)

[27] Rodrigues, S.B., Collinson, D.L.: 'having fun'?: Humour as resistance in brazil. Organization studies **16**(5), 739–768 (1995)

[28] Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake News Detection on Social Media: A Data Mining Perspective. SIGKDD Explor. **19**(1), 22–36 (Sep 2017). 10.4114/intartif.vol26iss71pp114-130: 10.1145/3137597.3137600, `https://dl.acm.org/doi/10.1145/3137597.3137600`

[29] Cardoso Durier da Silva, F., Vieira, R., Garcia, A.C.: Can machines learn to detect fake news? a survey focused on social media. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019)

[30] Silva, R.M., Santos, R.L., Almeida, T.A., Pardo, T.A.: Towards automatically filtering fake news in portuguese. Expert Systems with Applications **146**, 113199 (2020). 10.4114/intartif.vol26iss71pp114-130: https://doi.org/10.1016/j.eswa.2020.113199, `http://www.sciencedirect.com/science/article/pii/S0957417420300257`

[31] Suhaimin, M.S.M., Hijazi, M.H.A., Alfred, R., Coenen, F.: Natural language processing based features for sarcasm detection: An investigation using bilingual social media texts. In: 2017 8th International conference on information technology (ICIT). pp. 703–709. IEEE (2017)

[32] Wang, J., Fu, J., Xu, Y., Mei, T.: Beyond object recognition: Visual sentiment analysis with deep coupled adjective and noun neural networks. In: IJCAI. pp. 3484–3490 (2016)

[33] Wang, L., Wang, Y., De Melo, G., Weikum, G.: Five shades of untruth: Finer-grained classification of fake news. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 593–594. IEEE (2018)