Anomaly Detection in Online User Review

Ananthula Jeevan Kumar, Attipamula Anand Netha,
Cherukuri Jaswanth Santosh Kumar, Bandi Pavan Kalyan Reddy
and Ankireddypalle Shashank Kumar Reddy

April 6, 2024

Anomaly Detection in online user review

A.Jeevan
9921004828@klu.ac.in

A.Anand
99210041783@klu.ac.in

Ch.J.Santhosh kumar
9921004135@klu.ac.in

B.Pavan Kalyan reddy
9921004838@klu.ac.in

A.Shashank Kumar
99210041137@klu.ac.in

Abstract

This paper presents a pipeline to detect and explain anomalous reviews in online platforms. The pipeline is made up of three modules and allows the detection of reviews that do not generate value for users due to either worthless or malicious composition. The classifications are accompanied by a normality score and an explanation that justifies the decision made. The pipeline's ability to solve the anomaly detection task was evaluated using different datasets created from a large Amazon database. Additionally, a study comparing three explainability techniques involving 241 participants was conducted to assess the explainability module. The study aimed to measure the impact of explanations on the respondents' ability to reproduce the classification model and their perceived usefulness. This work can be useful to automate tasks in review online platforms, such as those for electronic commerce, and offers inspiration for addressing similar problems in the field of anomaly detection in textual data. We also consider it interesting to have carried out a human evaluation of the capacity of different explainability techniques in a real and infrequent scenario such as the detection of anomalous reviews, as well as to reflect on whether it is possible to explain tasks as humanly subjective as this one.

Keywords: Anomaly Detection, Text Reviews, Transformers, Explainability

1. Introduction

Nowadays more than ever in history, the user opinions about products and services have a great impact on the future of the company that offers them. In such a globalized and highly competitive world, online review platforms, such as electronic commerce (e-commerce), play a crucial role in the credibility of products and services. These user reviews usually come in the form of text reviews or numerical ratings, accompanied in some cases by images or videos. These reviews provide users with information about the product or service they are considering purchasing, which directly influences the number of sales. Most people make purchase decisions based on ratings and reviews from other users [1].

In the case of many companies, such as Amazon, each product in the store has a list of text reviews published by customers of the platform. Users can access this list of reviews (opinions) to obtain extra information about the product, being able to mark the reviews as useful, which will position those reviews with the most votes at the top of the list. In addition to this, users can report to Amazon if they feel a review is inappropriate, for example, if its content is incorrect. This procedure to rank reviews based on their usefulness and report inappropriate reviews is carried out manually by platform users. As a result, Amazon reported more than 200 million suspected fake reviews in 2020 alone [2]. This problem does not only occur on Amazon, but affects all platforms that allow their users to post reviews. For example, Tripadvisor uses an automatic system capable of distinguishing between normal, suspicious and inappropriate reviews [3]. Inappropriate ones are automatically removed (3.1% of review submissions in 2020), while those classified as suspicious are reviewed again by a human moderator (5.1% of review submissions in 2020).

On the other hand, in machine learning (ML), anomaly detection (AD) is the branch that builds models capable of differentiating between normal and abnormal data [4]. At first, anomaly detection might seem like a classification problem with only two classes. However, anomalies tend to occur infrequently or are non-existent, so normal data prevails in these scenarios. Because of this, it is common for models to be trained only with normal data. The goal of these models is to represent the normal class as well as possible in order to classify new data as normal or anomalous.

The technological development of recent years has allowed the construction of very powerful models for Natural Language Processing (NLP) [5].

Contrary to other tasks such as Sentiment Analysis [6] or Question Answering [7], the application of anomaly detection on texts is still at an early stage, probably due to its lower demand. For this reason, we present in this article a pipeline that, given text reviews of a product (in this case from Amazon), addresses opinion filtering as an anomaly detection problem where:

- Reviews containing representative information about the product are considered as the normal class.
- Reviews whose content has nothing to do with the product to be represented are considered as the anomalous class.

The proposed pipeline allows us to carry out the following tasks:

- Classify reviews as normal or anomalous, allowing us to locate those that do not describe characteristics of the product to which they are associated, and therefore have no value for the users of the platform.
- Issue a normality score associated with each review.
- Generate an explanation that justifies the classification made for each review by the system.

The pipeline's ability to solve the anomaly detection task was evaluated using different datasets created from a large Amazon database [8]. In addition, to evaluate the explainability module, a study was carried out to compare three explainability techniques in which a total of 241 people had participated. The objective of this study is both to measure the impact of the explanations on the reproducibility of the classification model by the respondents, as well as the usefulness of these explanations.

We believe that this work can be useful to automate tasks such as those mentioned in online review platforms, in addition to the existing interest in the application of anomaly detection models on texts, for which it can serve as inspiration to solve similar problems. We also consider it interesting to have carried out a human evaluation of the capacity of different explainability techniques in a real and infrequent scenario such as the detection of anomalous reviews, as well as to reflect on whether it is possible to explain tasks as humanly subjective as this one.

This document is structured as follows. Section 2 contains a brief review of the main anomaly detection works in texts and, more specifically, in text reviews. Section 3 describes the proposed pipeline and its operation. Section 4 collects the experimentation carried out and, finally, conclusions are drawn in Section 5.

2. Related work

Anomaly detection has been a consolidated research field for years. Its great utility has allowed its techniques to be applied in numerous areas: medicine [9], industrial systems [10], electronic fraud [11], cybersecurity [12], etc. However, when we talk about texts and NLP, there is no massive application of these anomaly detection techniques as in the previous cases. This may be due to the difficulty in defining the idea of an anomaly in texts. Contrary to other scenarios, such as monitoring an industrial system through its sensors, in which the anomalous class will correspond to faults in the system, when the data is text, defining the concept of an anomaly is not trivial.

One of the most important lines of research related to the detection of anomalies in texts is fake reviews detection, also known as spam review detection, fake opinion detection, and spam opinion detection [13]. The main problem associated with fake review detection is classifying the review as either fake or genuine. There are generally three types of fake reviews [14]:

- **Type 1 (untruthful opinions)**: Fake reviews describing users who post negative reviews to damage a product's reputation or post positive reviews to promote it. These reviews are called fake or deceptive reviews, and they are difficult to detect simply by reading, as real and fake reviews are similar to each other.

- **Type 2 (reviews on brands only)**: Those that do not comment on the products themselves, but talk about the brands, manufacturers, or sellers of the products. Although they can be useful, they are sometimes considered spam because they are not targeted at specific products.

- **Type 3 (non-reviews)**: Non-reviews that are irrelevant and offer no genuine opinion.

In the work carried out by J. Salminen *et al.* [15], the authors try to distinguish genuine reviews from fake reviews on Amazon. To have a labeled dataset of fake reviews, they use GPT-2 [16] to artificially generate them. After this, they solve the task of distinguishing between genuine and fake reviews by fine-tuning a pretrained RoBERTa [17] model. They also show that his model is also capable of detecting fake reviews manually written by humans. Ş. Öztürk Birim *et al.* [18] proposed to use relevant information as the review length, purchase verification, sentiment score, or topic distribution as features to represent costumers reviews. Based on these features, well-known machine learning classifiers like random forests (RF) [19] are applied for fake detection. In their article, D. U. Vidanagama *et al*. [20] incorporate review-related features such as linguistic features, Part-of-Speech (POS) features, and sentiment analysis features using a domain ontology to detect fake reviews with a rule-based classifier.

L. Ruff *et al*. [21] presented Context Vector Data Description (CVDD), a text-specific anomaly detection method that allows working with sequences of variable-length embeddings using self-attention mechanisms. To overcome the limitations of CVDD, J. Mu *et al*. [22] proposed tadnet, a textual anomaly detection network that uses an adversarial training strategy to detect anomalous texts in Social Internet of Things. In addition, thanks to the capture of the different semantic contexts of the texts, both models achieve interpretability and flexibility, allowing to detect which parts of the texts have caused the anomaly.

Other authors, instead of developing text-specific AD models, make use of well-known AD and NLP techniques to design architectures that solve the problem. B. Song *et al*. [23] propose to analyze the accident reports of a chemical processing plant to detect anomalous conditions. In this scenario, anomalous conditions are defined as unexperienced accidents that occur in unusual conditions. The authors work directly with the original text extracting the meaningful keywords of the reports using the term frequency-inverse document frequency (TF-IDF) index. Based on this, and using the local outlier factor (LOF) algorithm, they identify anomaly accidents in terms of local density clusters, finding four major types of anomaly accidents. Working with the original texts and not with embeddings they achieve a certain interpretability in the results. In the approach presented by S. Seo *et al.* [24], a framework for identifying unusual but noteworthy customer responses and extracting significant words and phrases is proposed. The authors use Doc2Vec to vectorize customer responses, then LOF is applied to identify unusual responses, and based on a TF-IDF analysis and the distances in the embedding space, they can visualize useful information about the results through a network graph.

In the previous works and most cases, the detection of fake reviews focuses

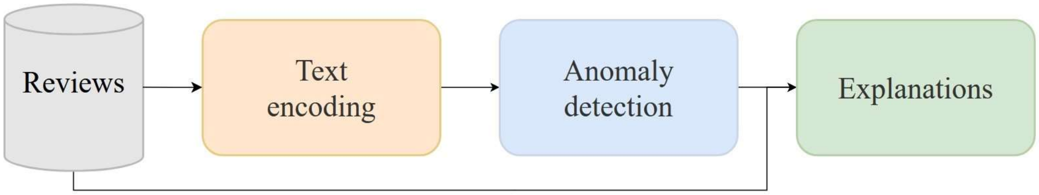


Figure 1: General modules that form the proposed pipeline.

on detecting type 1 reviews, that is, reviews that positively or negatively describe a product but whose intention is not genuine since they do not come from a real buyer. Since there are several works based on this scenario [15, 18, 20], in this work we will focus on detecting type 2 and 3 reviews, which refer to reviews that do not provide information about the product itself. In this way, the objective of this work is to design a pipeline capable of distinguishing reviews related to a specific product (normal reviews) from reviews that do not and therefore do not provide information to the users that read them (anomalous reviews), for example, because they wrongly describe other products or because they are too generic. In addition to this, the classifications carried out by the system will be explained through an analysis process based on NLP techniques.

3. The proposed pipeline

The purpose of the proposed pipeline is, given the text reviews of an Amazon product, to classify them as normal if they refer to it, and as anomalous if they describe different products or if they are so generic that they do not provide useful information to consumers. These classifications will be accompanied by a normality score and explanations that justify them. Figure 1 shows the modules that are part of the proposed pipeline to solve this task. In the first phase, we encode the text reviews of the target product using the pretrained MPNet transformer [25]. In the second phase, a DAEF [26] autoencoder is trained using these embeddings. Using the reconstruction errors issued by the network and a predefined threshold error, we can classify reviews as normal (error greater than the threshold) or anomalous (error less than the threshold). For the last phase, we propose a method based on the most frequent normal terms to generate an explanation associated with each classification.

Therefore, the proposed pipeline is composed of three main modules: (1) Text encoding; (2) Anomaly detection; (3) Explainability. The following sections describe each of them in detail.

3.1. Text encoding

Although there are models capable of dealing directly with images or text, machine learning models are usually designed to be trained using numerical vectors that represent the sample data (tabular data). This input format of the data is the usual one for classic anomaly detection methods. In the NLP area, there are multiple techniques to represent texts using vectors of real numbers, which are known as embeddings. These techniques allow the generation of vector spaces that try to represent the relationships and semantic similarities of the language, so that, for example, two synonymous words will be found at a shorter distance in the vector space than two unrelated words.

Embeddings can be calculated independently for each word of the language (word embeddings), which led to models such as word2vec [27] or Global Vectors (GloVe) [28]. The representation of a sentence (sentence embedding) or a document (document embedding) will therefore be the sum of all the individual representations of the terms that make it up. To obtain representations of fixed length, it is usual to perform operations such as the mean. In certain cases, these operations between embeddings can worsen or even invalidate the final embedding, so specific models have been developed capable of understanding and representing a text as a whole, instead of just encoding it word by word. Among these models are those based on transformers [29], such as BERT [30], XLNet [31], GPT-3 [32] or GPT-4 [33], which have been trained over large-scale datasets and can solve different tasks, including sentence embedding.

Since it inherits the advantages of the BERT and XLNet models while overcoming their limitations, we will use the pre-trained MPNet [25] model to calculate the embeddings of the reviews. MPNet combines Masked Language Modeling (MLM) and Permuted Language Modeling (PLM) to predict token dependencies, using auxiliary position information as input to enable the model to view a complete sentence and reduce position differences. Models like GPT-3 or GPT-4 are more advanced, but in addition to not being open source, they demand much higher computational resources, unaffordable for a significant part of the scientific community, as is our particular case.

MPNet maps sentences and paragraphs to a 768 dimensional dense vector space $\mathbf{e} \in \mathsf{R}^{768\times 1}$, providing fast and quality encodings. Computing time is a critical aspect in the area in which this work is framed, since in e-commerce platforms (and online reviews in general) we can find a huge number of products and reviews to deal with. Therefore, the proposed pipeline uses MPNet to calculate the embeddings of the reviews, which allows them to be represented as numeric vectors of fixed size. This model and others are available in the Hugging Face repository [34].

3.2. Anomaly detection

Anomaly detection is a field with a large number of algorithms that solve the problem of distinguishing between normal and anomalous instances in a wide variety of ways [4, 35]. Depending on the assumptions made and the processes they employ, we can distinguish between five main types of methods: probabilistic, distance-based, information theory-based, boundary-based, and reconstruction-based methods. Among the latter are autoencoder networks (AE) [36], one of the most widely used models. AE are a type of self-associative neural network whose output layer seeks to reproduce the data presented to the input layer after having gone through a dimensional compression phase. In this way, they manage to obtain a representation of the input data in a space with a dimension smaller than the original, learning a compact representation of the data, retaining the important information, and compressing the redundant one.

One of the requirements of our proposed pipeline is to generate a normality score for each review. This score will allow us, among other things, to order the different reviews by their level of normality. When AE networks are used in anomaly detection scenarios, the classification is usually carried out based on the reconstruction error that they emit to reproduce in its output the embeddings of the reviews it receives as inputs, which represents the level of normality of the evaluated instance. This reconstruction error can be used as the normality score we are looking for.

Due to the speed of its training, we have decided to use DAEF (Deep Autoencoder for Federated learning) [26] as our anomaly detection model. Unlike traditional neural networks, DAEF trains a deep AE network in a non-iterative way, which drastically reduces its training time without losing the performance of traditional (iterative) AEs. The proposed pipeline uses DAEF to calculate the embeddings of the reviews, being able to issue a normality score associated with each of the review classifications.

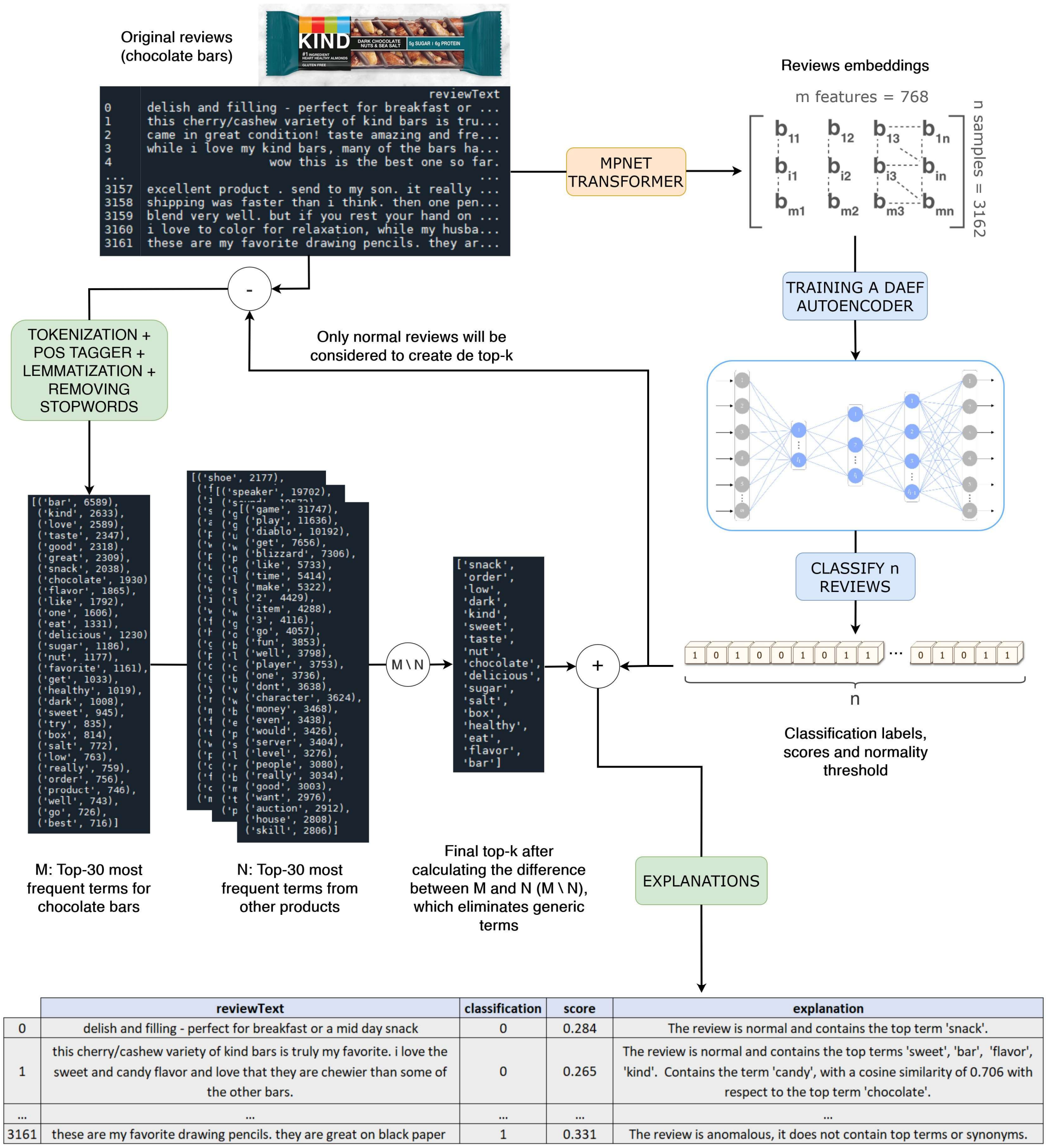
3.3. Explainability

As black-box ML models are mainly being employed to make important predictions in critical contexts, the demand for transparency is increasing. The danger is in creating and using decisions that are not justifiable, legitimate, or that do not allow obtaining detailed explanations of their behaviour. Explanations supporting the output of a model are crucial, moreover in fields such as medicine, autonomous vehicles, security, or finance.

When ML models do not meet any of the criteria imposed to declare them explainable, a separate method must be applied to the model to explain its decisions. This is the purpose of post-hoc explainability techniques [37], which aim at communicating understandable information about how an already developed model produces its predictions for any given input. Within these techniques, the so-called model-agnostic techniques are those designed to be plugged into any model with the intent of extracting some information from its prediction procedure.

To generate the explanations we have implemented an approach based on a statistical analysis of the dataset. Based on the definitions of normal and anomalous review presented in Section 2, Our hypothesis assumes that normal reviews will always refer directly or indirectly to the target product so that there will be a list of terms used very frequently among normal texts. The appearance of one or more of these "normal" terms in a review would justify its classification by the system as normal. In the same way, anomalous reviews may be justified with the non-presence of said terms.

The original text reviews classified as normal using the AD model will be processed (tokenization, lemmatization and stopword removal) and analyzed to obtain the list of the top-n most frequent terms. This list of terms will be compared with the lists of frequent terms of other products in order to remove the terms they have in common, so that the final list for a product does not contain generic terms. This final list of terms will be used to generate the explanations. New reviews classified as normal will be searched for the presence of any of these terms. To make this search more flexible, it will also be analyzed if there are terms in the review semantically close to any of the words on the list. To quantify the closeness of two words, the cosine similarity emitted by the MPNet transformer model itself will be used as a measure of distance. Reviews classified as anomalous will be explained by the non-occurrence of terms from the list. Figure 2 represents an overall view of the pipeline considering the product "chocolate bars" as the normal class.


Original reviews
(chocolate bars)
KIND
DARK CHOCOLATE NUTS & SEA SALT
5g SUGAR | 6g PROTEIN
#1 INGREDIENT HEART HEALTHY ALMONDS
GLUTEN FREE
MPNET TRANSFORMER
Reviews embeddings
m features = 768
n samples = 3162
b11 b12 b13 b1n
bi1 bi2 bi3 bin
bm1 bm2 bm3 bmn
-
TOKENIZATION + POS TAGGER + LEMMATIZATION + REMOVING STOPWORDS
Only normal reviews will be considered to create de top-k
TRAINING A DAEF AUTOENCODER
CLASSIFY n REVIEWS
M \ N
+
1 0 1 0 0 1 0 1 1 ... 0 1 0 1 1
n
Classification labels, scores and normality threshold
M: Top-30 most frequent terms for chocolate bars
N: Top-30 most frequent terms from other products
Final top-k after calculating the difference between M and N (M \ N), which eliminates generic terms
EXPLANATIONS

```
                                             reviewText
0     delish and filling - perfect for breakfast or ...
1     this cherry/cashew variety of kind bars is tru...
2     came in great condition! taste amazing and fre...
3     while i love my kind bars, many of the bars ha...
4                        wow this is the best one so far.
...                                                   ...
3157  excellent product . send to my son. it really ...
3158  shipping was faster than i think. then one pen...
3159  blend very well. but if you rest your hand on ...
3160  i love to color for relaxation, while my husba...
3161  these are my favorite drawing pencils. they ar...
```

```
[('bar', 6589),
 ('kind', 2633),
 ('love', 2589),
 ('taste', 2347),
 ('good', 2318),
 ('great', 2309),
 ('snack', 2038),
 ('chocolate', 1930)
 ('flavor', 1865),
 ('like', 1792),
 ('one', 1606),
 ('eat', 1331),
 ('delicious', 1230)
 ('sugar', 1186),
 ('nut', 1177),
 ('favorite', 1161),
 ('get', 1033),
 ('healthy', 1019),
 ('dark', 1008),
 ('sweet', 945),
 ('try', 835),
 ('box', 814),
 ('salt', 772),
 ('low', 763),
 ('really', 759),
 ('order', 756),
 ('product', 746),
 ('well', 743),
 ('go', 726),
 ('best', 716)]
```

```
[('shoe', 2177),
```

```
[('speaker', 19702),
```

```
[('game', 31747),
 ('play', 11636),
 ('diablo', 10192),
 ('get', 7656),
 ('blizzard', 7306),
 ('like', 5733),
 ('time', 5414),
 ('make', 5322),
 ('2', 4429),
 ('item', 4288),
 ('3', 4116),
 ('go', 4057),
 ('fun', 3853),
 ('well', 3798),
 ('player', 3753),
 ('one', 3736),
 ('dont', 3638),
 ('character', 3624),
 ('money', 3468),
 ('even', 3438),
 ('would', 3426),
 ('server', 3404),
 ('level', 3276),
 ('people', 3080),
 ('really', 3034),
 ('good', 3003),
 ('want', 2976),
 ('auction', 2912),
 ('house', 2808),
 ('skill', 2806)]
```

```
['snack',
 'order',
 'low',
 'dark',
 'kind',
 'sweet',
 'taste',
 'nut',
 'chocolate',
 'delicious',
 'sugar',
 'salt',
 'box',
 'healthy',
 'eat',
 'flavor',
 'bar']
```

	reviewText	classification	score	explanation
0	delish and filling - perfect for breakfast or a mid day snack	0	0.284	The review is normal and contains the top term 'snack'.
1	this cherry/cashew variety of kind bars is truly my favorite. i love the sweet and candy flavor and love that they are chewier than some of the other bars.	0	0.265	The review is normal and contains the top terms 'sweet', 'bar', 'flavor', 'kind'. Contains the term 'candy', with a cosine similarity of 0.706 with respect to the top term 'chocolate'.
...	...	...	...	...
3161	these are my favorite drawing pencils. they are great on black paper	1	0.331	The review is anomalous, it does not contain top terms or synonyms.

Final output with explanations

Figure 2: Proposed pipeline considering the product “chocolate bars” as the normal class.

In summary, in this work, we propose the use of this statistical analysis of the terms to generate the explanations associated with the classification of the reviews.

4. Evaluation

In this section several experiments are presented to show the behavior of the proposed pipeline in real scenarios. The section is divided into two parts: the evaluation of the anomaly detection task (Section 4.1), and the evaluation of explanations (Section 4.2). In the first one, the capability of the pipeline to detect anomalous reviews will be evaluated using products from the Amazon platform. The second part will discuss the problems of evaluating explanations and will present a study involving humans to assess the benefits of adopting such explanations.

4.1. Evaluating the anomaly detection task

The anomaly detection task that solves the proposed pipeline can be evaluated following the usual methodology in the field of anomaly detection. In this first part of the evaluation we will describe the test scenario used, the methodology employed, and we will discuss the results obtained.

4.1.1. Experimental setup

The objective of this study is to evaluate the capacity of the pipeline in a real anomaly detection scenario. Although Section 3.2 defines the requirements for the anomaly detection methods that the pipeline can use, in these tests different alternatives have been compared. Specifically, two non-iterative implementations of autoencoder networks and a boundary-based method have been employed. These methods are DAEF [26], OS-ELM [38], and OC-SVM [39] respectively.

For the evaluation, we employed datasets obtained from the large Amazon database [8], which collects reviews of various products from the year 1996 to 2018. The main problem with this dataset is that the reviews that compose it are not labeled as normal or anomalous. Because of this, in order to simulate a scenario similar to the one described throughout the article, we have decided to select the reviews corresponding to several of the most demanded products. Thus, for a given test we could consider the reviews of one product as normal, and introduce reviews from other products as anomalous. Five different product categories were selected, for which the two products with the highest

Table 1: Characteristics of the products used.

Product	Category	Reviews
Chocolate bars	Grocery and Gourmet Food	11526
Anise seeds	Grocery and Gourmet Food	9083
Colored pencils	Office Products	14340
Ergonomic cushion	Office Products	11942
Gaming mouse	Video Games	6462
PS4 membership	Video Games	5135
Bluetooth speaker	Electronics	28539
Wi-Fi range extender	Electronics	20873
Foot insoles	Amazon Fashion	4384
Yoga leggings	Amazon Fashion	3889

number of reviews were used, resulting in a total of 10 products. Table 1 summarizes its characteristics. In all cases, MPNet was used as the model to encode the text reviews.

Besides we considered two types of tests based on the products used:

- **1vs.4 - Far products**: For each of the five categories, in this type of tests the product with the most reviews from one of the categories will be considered as the normal class, while the product with the highest number of reviews from each of the other four categories will be considered as the anomalous class. The fact that the product considered normal belongs to a different category should facilitate its distinction.

- **1vs.1 - Near products**: For each of the five categories, in these tests the two products with the most reviews of the same category will be selected. One will be considered as the normal class and the other as the abnormal class. The fact that both products belong to the same category should make it more difficult to distinguish them since they may have common characteristics.

The anomaly detection algorithms were trained using only normal data (the product considered as normal), while the test phase included data from both classes in a balanced manner (50% normal and 50% anomalies).

To evaluate the performance of each algorithm with each combination of hyperparameters a 10-fold has been used. The normal data were divided into 10-folds so that, at each training run, 9 folds of normal data were used for training, while the remaining fold and the anomaly set were used together in a balanced manner for testing.

In this work, reconstruction errors at the output of the anomaly detector were calculated using Mean Squared Error (MSE). To establish the threshold above which this error indicates a given instance corresponds to an anomaly, among the various methods available, we employed a popular approach based on the interquartile range (IQR) of the reconstruction errors of the training examples, defined by:

$$IQR = Q_3 - Q_1 \tag{1}$$

where Q_1 and Q_3 represent the first and the third quartiles. We define two error thresholds, one for outliers (*outlierIQR*) and another for extreme outliers (*extremeIQR*), as:

$$outlierIQR = Q_3 + 1.5 \times IQR \tag{2}$$

$$extremeIQR = Q_3 + 3 \times IQR \tag{3}$$

In addition, throughout the tests, we also considered other thresholds using fixed percentiles (Q_{95}, Q_{90}, Q_{80}, Q_{70}, Q_{60} and Q_{50}), since a priori it is not easy to figure out which one can provide the best results, so it is considered as an additional hyperparameter to be taken into account. The OC-SVM method constitutes an exception as it automatically assigns a score to each input instance and decides its classification based on an internally calculated threshold error.

To measure the performance of the algorithms the F1-score metric was used, considering the anomalous class as the positive one, and based on the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN):

$$F_1 = 2\,\frac{precision \cdot recall}{precision + recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{4}$$

Finally, the combinations of hyperparameters chosen for each algorithm, as well as the error thresholds, were selected using a grid search and are available in Appendix A (Tables A.9 and A.10).

All the evaluation tests were performed in a machine equipped with an Intel Core i7-11700k processor and 64GB of RAM.

4.1.2. Evaluating the anomaly detection task

Table 2 shows the results of the 1vs.4 - Far products test. As can be seen, the performance of the three models is remarkable, highlighting that obtained by the OS-ELM autoencoder network. We can affirm that the embeddings produced by the MPNet model provide an encoding with sufficient quality for the anomaly detection models to be able to differentiate between the evaluated products.

Table 2: Average test F1-score ± standard deviation for the 1vs.4 datasets.

Normal class	Anomalous class	DAEF	OS-ELM	OC-SVM
Chocolate bars	Colored pencils, Gaming mouse, Bluetooth speaker, Foot insoles	92.1±1.0	96.6±0.4	95.3±1.5
Colored pencils	Chocolate bars, Gaming mouse, Bluetooth speaker, Foot insoles	96.1±0.4	96.0±0.3	94.4±1.1
Gaming mouse	Chocolate bars, Colored pencils, Bluetooth speaker, Foot insoles	94.5±1.1	95.7±0.1	93.8±0.8
Bluetooth speaker	Chocolate bars, Colored pencils, Gaming mouse, Foot insoles	95.7±0.8	96.9±0.2	94.6±0.3
Foot insoles	Chocolate bars, Colored pencils, Gaming mouse, Bluetooth speaker	96.4±1.3	96.3±0.2	94.5±0.6

Table 3 collects the results of test 1vs.1 - Near products. As can be seen, the overall performance is still good, although it has been slightly reduced concerning the previous test, possibly because the task is a little more complicated as the products are closer together. Nevertheless, once again, the quality of the embeddings allows a proper differentiation between products.

Table 3: Average test F1-score ± standard deviation for the 1vs.1 datasets.

Normal class	Anomalous class	DAEF	OS-ELM	OC-SVM
Chocolate bars	Anise seeds	91.4±2.1	91.8±0.1	90.7±1.3
Anise seeds	Chocolate bars	90.3±0.9	91.4±0.3	90.0±0.5
Colored pencils	Ergonomic cushion	96.5±1.1	96.4±0.4	94.6±1.1
Ergonomic cushion	Colored pencils	96.6±0.7	96.1±0.1	94.8±0.8
Gaming mouse	PS4 membership	93.8±1.4	94.1±0.3	91.9±0.3
PS4 membership	Gaming mouse	92.0±1.1	94.0±0.2	92.1±0.5
Bluetooth speaker	Wi-Fi range extender	90.2±3.1	89.4±0.6	90.0±0.7
Wi-Fi range extender	Bluetooth speaker	93.0±0.5	92.9±0.2	92.3±0.4
Foot insoles	Yoga leggings	90.4±1.0	92.2±0.2	90.5±0.4
Yoga leggings	Foot insoles	89.9±2.2	90.9±0.3	90.4±0.6

4.2. Evaluating the explanations for model classifications

In Section 3.3, we proposed as a method of explanation to support the model decisions one based on the occurrence of frequent terms, based on the hypothesis that normal reviews will tend to use certain terms regularly, while anomalous reviews will not. In this section, this approach is compared, qualitatively through user surveys, with two other popular alternative approaches to achieve such explainability, specifically SHAP [40] and GPT-3 [32].

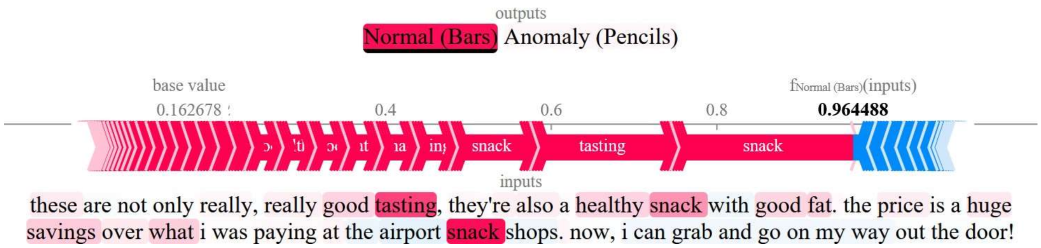
4.2.1. Explanations based on SHAP

Many explainability techniques base their operation on determining which characteristics of the dataset have most influenced the predictions. For example, in industrial scenarios, it is common for the features of the datasets to come directly from the physical aspects measured by the sensors of the machines, giving rise to variables such as temperatures, pressures or vibrations. By quantifying the influence of each of these variables on the output of the system we can achieve very useful explanations.

However, in the scenario proposed in this paper, the data received by the anomaly detection model as input are the reviews' embeddings, these being numerical vectors. The variables that make up the embeddings are not associated with aspects understandable by human beings such as temperatures or pressures, so determining which ones have influenced the most would not provide us with useful information.

To solve this problem, some explainability techniques have been adapted to deal with texts. In the case of NLP models, specifically transformers, SHAP (SHapley Additive exPlanations) [40] is one of the most widely used techniques. SHAP is a game theory-based approach to explain the output of any machine learning model. SHAP also assigns each feature an importance value for a particular prediction, but it has been extended to provide interpretability to models that use embeddings as input. In these cases, it can quantify the importance of each word of the original text in the prediction, which generates a quite understandable interpretation for a human being.

In this work, we considered that SHAP could be a good alternative to our proposal to generate the explanations associated with the classification of the reviews. Figure 3 shows the explanation generated by SHAP for a review classified as normal, where it can be seen that each term of the review has an associated score representing its influence on the classification. Despite this, for the evaluation of explanation techniques and to facilitate the understanding of the explanation by the final users, during this work (many of them



these are not only really, really good tasting, they're also a healthy snack with good fat. the price is a huge savings over what i was paying at the airport snack shops. now, i can grab and go on my way out the door!

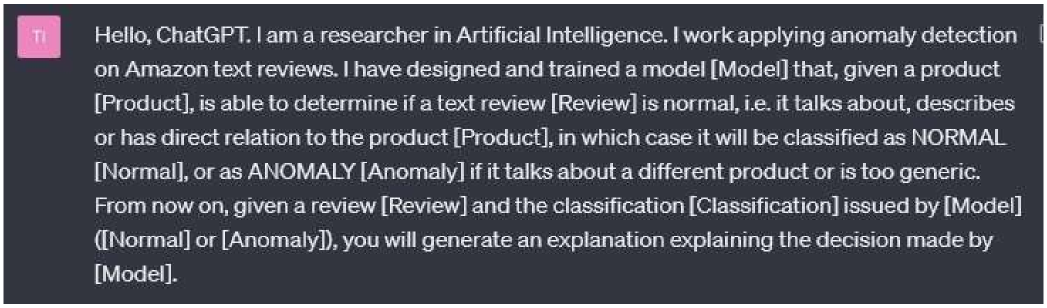
Figure 3: Explanation generated by SHAP for the anomalous reviews detection problem raised in this work. In this scenario, the reviews to be analysed correspond to the product "chocolate bars" (Bars). The review of this example was correctly classified as normal. The most influential terms in its classification as normal are marked in red, while the terms that promote the opposite class are highlighted in blue, in this case practically none since we are dealing with an obvious case. Greater intensity implies greater influence. In this case, the review terms that have most influenced its classification as normal are "snack" and "tasting".

unfamiliar with these techniques), the explanations generated by SHAP will be simplified, showing only the five most influential terms, instead of all.

4.2.2. Explanations based on GPT-3

There is no doubt that GPT-3 has started a technological revolution at all levels. Its availability to the general public has led to the discovery of a large number of unimaginable features before its release. The original idea was to create a high-level conversational bot, trained with a large amount of text available on the web, such as books, online encyclopedias or forums. However, its deep understanding of language has far exceeded the preset idea of a chatbot. GPT-3 is capable of successfully carrying out tasks that go beyond writing a joke or summarizing a novel, GPT-3 is capable of analyzing and developing code in multiple programming languages, generating SEO positioning strategies, or carrying out NLP tasks traditionally solved by ad hoc models, such as sentiment analysis.

Due to its enormous potential, in this work, we have decided to study GPT-3 as an explainability model. To do this, first of all, we have introduced a prompt in which we describe the task that our anomaly detection model is carrying out, as well as the format with which we will work in future prompts (see Figure 4). After this, and following the predefined format, GPT-3 is ready to generate the explanations (see Figure 5). Although GPT-3 does not have direct knowledge of the anomaly detection model, its ability to generate consistent and intuitive responses can be of great help to humans



Hello, ChatGPT. I am a researcher in Artificial Intelligence. I work applying anomaly detection on Amazon text reviews. I have designed and trained a model [Model] that, given a product [Product], is able to determine if a text review [Review] is normal, i.e. it talks about, describes or has direct relation to the product [Product], in which case it will be classified as NORMAL [Normal], or as ANOMALY [Anomaly] if it talks about a different product or is too generic. From now on, given a review [Review] and the classification [Classification] issued by [Model] ([Normal] or [Anomaly]), you will generate an explanation explaining the decision made by [Model].

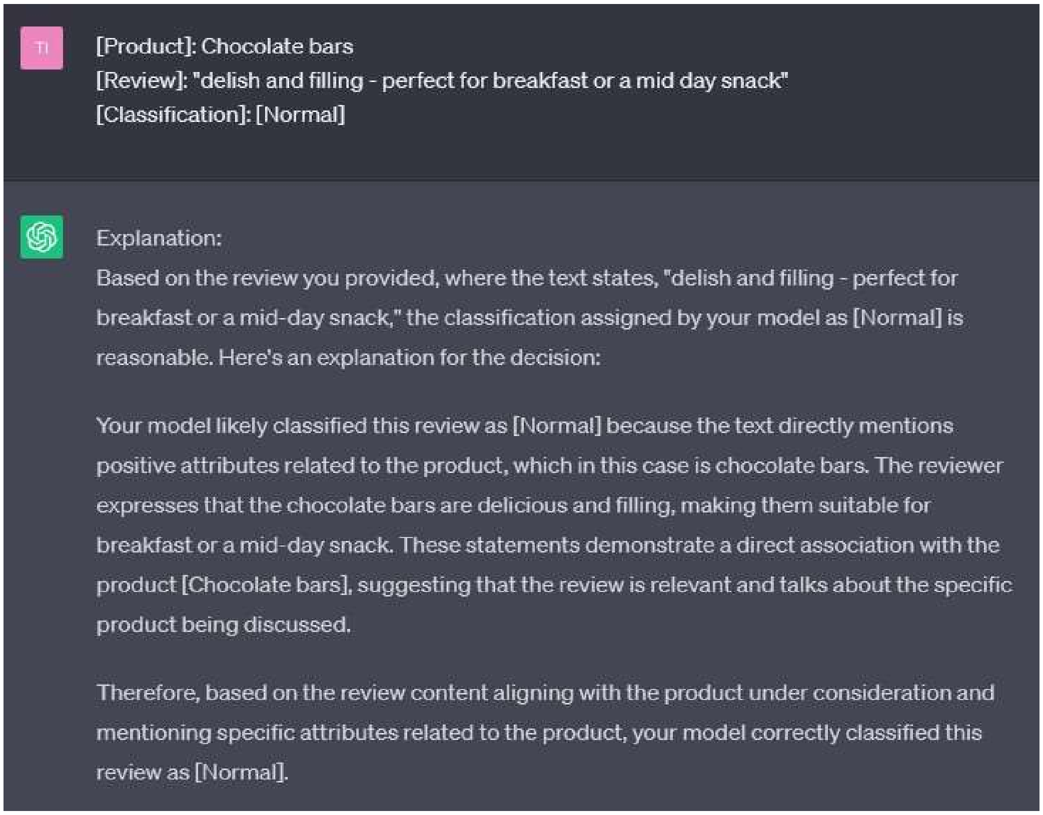
Figure 4: Initial prompt to present the anomaly detection problem to be solved to GPT-3.

reading the explanations.

We should note that since GPT-3 trains on a wide range of data from the internet, including data that may contain biases, there is a risk that the generated explanations may reflect or amplify those biases. It is essential to exercise caution and perform a critical analysis of the explanations generated, considering their context and possible inherent biases.

4.2.3. How to evaluate explainability

Unlike other tasks such as anomaly detection, the quality of the explanations generated to provide certain interpretability to a model is not easily measurable. In a scenario like the one described in this paper, the subjectivity of the users plays a strong role when determining whether an explanation is appropriate or not. Due to this, we have decided to carry out a comparative study of the three explainability techniques by means of a survey. This survey was disseminated through the students, professors, and research and administrative staff of our university, giving rise to a total of 241 participants. Table 4 shows the number of participants by area of knowledge. To build the survey, reviews from the “1 vs. 4” scenario described in Table 2 were used, considering “chocolate bars” as the normal class, and DAEF as an anomaly detection method.



[Product]: Chocolate bars
[Review]: "delish and filling - perfect for breakfast or a mid day snack"
[Classification]: [Normal]

Explanation:
Based on the review you provided, where the text states, "delish and filling - perfect for breakfast or a mid-day snack," the classification assigned by your model as [Normal] is reasonable. Here's an explanation for the decision:

Your model likely classified this review as [Normal] because the text directly mentions positive attributes related to the product, which in this case is chocolate bars. The reviewer expresses that the chocolate bars are delicious and filling, making them suitable for breakfast or a mid-day snack. These statements demonstrate a direct association with the product [Chocolate bars], suggesting that the review is relevant and talks about the specific product being discussed.

Therefore, based on the review content aligning with the product under consideration and mentioning specific attributes related to the product, your model correctly classified this review as [Normal].

Figure 5: Prompt in which the classification of a review is explained by GPT-3. The product considered as normal is chocolate bars and the review has been classified as normal by the anomaly detection model.

Table 4: Area of knowledge of the survey participants.

Knowledge area	Participants
Engineering and Architecture	87 (36.1%)
Social and Legal Sciences	77 (32.0%)
Natural Sciences	33 (13.7%)
Arts and Humanities	23 (9.5%)
Health Sciences	17 (7.1%)
Others	4 (1.7%)

The survey consists of two different tests: (1) Forward simulation [41], which allows measuring the effect of explanations on users; (2) Personal utility, which allows measuring the utility of the explanations based on the personal preferences of the users.

4.2.4. Forward simulation

This test is inspired by the work carried out by P. Hase et al. [41] and is divided into four phases: a Learning phase, a Pre-prediction phase, a Learning phase with explanations, and a Post-prediction phase. To begin, users are given 20 examples from the model's validation set with reviews and model predictions but no explanations. Then they must predict the model output for 10 new reviews. Users are not allowed to reference the learning data while in the prediction phases. Next, they return to the same learning examples, now with explanations included. Finally, they predict model behavior again on the same instances from the first prediction round. The classes of reviews chosen for each of the phases described are balanced between normal and abnormal. By design, any improvement in user performance in the Post prediction phase is attributable only to the addition of explanations that help the user to understand the model behaviour.

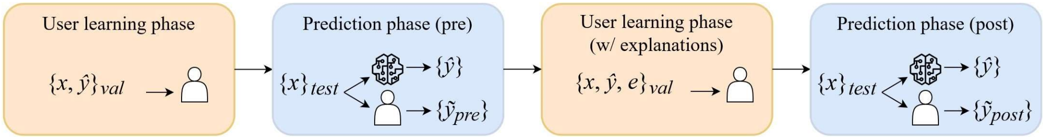


Figure 6: Forward simulation test procedure to measure human users' ability to understand and predict model behavior. To isolate the impact of explanations, baseline accuracy is measured first, followed by accuracy measurement after users have access to explanations of the model's behavior. The explained examples are different from the test instances.

Figure 6 represents this procedure, where x represents a review, $\hat{y}$ is the class predicted by the model, and $\tilde{y}$ the class predicted by the human simulation. Taking this into account, the Explanation Effect can be calculated as follows:

$$Explanation\, Effect = Post\ Accuracy - Pre\ Accuracy \tag{5}$$

where the pre and post accuracies area calculated comparing the user's prediction against the model's prediction. In order not to bias the results, we have decided that each person surveyed will only participate in the forward simulation of a single explainability technique, so that the total number of participants was divided randomly into three groups, each one associated with one of the three techniques: Terms frequency (78), SHAP (89), and GPT-3 (77). The three groups of participants will deal with the same reviews throughout the four phases of the test, only the explanations presented in the third phase of the test (learning with explanations) will vary depending on the assigned group/technique. Throughout the survey, we have warned users several times that they should try to simulate the behavior of the anomaly detection model, instead of ranking the reviews using their personal criteria.

Tables 5 and 6 show the results of this test. As can be seen in the first table, the average explanation effect of the three explainability techniques has not been very remarkable. The high standard deviation suggests a high variability between the different study participants, both in the initial (Pre) and subsequent (Post) classifications and therefore in the explanation effect. The initial difference between the groups in the Pre-accuracy and the closeness of the three techniques in the explanation effect does not allow us to opt for any of the three options.

In Table 6 we can see the results of the test based on the area of knowledge of the participants. The results obtained using the different explainability techniques have been aggregated. The purpose of this comparison is to analyze whether there is any relationship between the technical background of the respondents and their performance on the test. As can be seen, there are slight differences between the values of pre and post accuracy, with the group of respondents belonging to the area of natural sciences standing out, whose explanation effect was the only positive one (4.5%).

Table 5: Forward simulation tests. Average accuracy and Explanation effect (± standard deviation) for each explainability technique.

Technique	Pre Accuracy	Post Accuracy	Explanation Effect
Terms frequency	76.2 ±13.0	72.8±13.8	-3.4±12.7
SHAP	72.4±14.9	71.9±14.9	-0.5±13.8
GPT-3	69.7±15.2	70.1 ±17.1	0.4±14.3

Table 6: Forward simulation tests. Average accuracy and Explanation effect (± standard deviation) broken down by participants' area of knowledge.

Area of knowledge	Pre Accuracy	Post Accuracy	Explanation Effect
Engineering and Architecture	76.3 ±14.5	73.9±16.1	-2.4±12.9
Social and Legal Sciences	72.2±13.0	69.5±15.8	-2.7±13.0
Natural Sciences	70.9±14.7	75.4 ±12.8	4.5±13.5
Arts and Humanities	68.7±17.7	67.8 ±14.8	-0.9±19.0
Health Sciences	72.0±14.2	71.3 ±10.6	-0.7±10.3

Analyzing the results we can affirm that the initial accuracy (pre-accuracy) is quite high, which indicates that the respondents tend to successfully reproduce the behavior of the model even if they do not have the explanations. This could be because, in this case, both the input data to the AD model (in natural language) and the problem it solves are easily understandable to a human, which means that the respondents are able to solve the classification problem by themselves. The standard deviation accompanying the pre-accuracy results is notable but does not become too high, which reaffirms the previous argument.

After supplying the respondents with the explanations associated with the reviews, the post-accuracy obtained by them presents values very similar to the previous scores (pre-accuracy). We can therefore affirm that in general terms the effect of the explanations, in this case, has not been beneficial for the users during this test.

The non-improvement may be due to several reasons. One of them may be the presence of reviews that show a certain degree of ambiguity, which not only makes their classification difficult for the respondents, but also for the AD models. However, in real scenarios the occurrence of reviews whose normality score is around the threshold value would be something to be expected, not all events are easily classifiable. Another possible reason may be that the tendency of some users throughout the survey has been to classify the reviews using their personal criteria, rather than trying to simulate the behavior of the anomaly detection model.

4.2.5. Personal utility

After completing the first test, the participants were given a second exercise, common to all participants, regardless of the group to which they were assigned in the previous phase. This consists of a subjective evaluation of the three explainability techniques. The idea is to present the participant with a review, its classification by the model, and an explanation generated by each of the three explainability techniques. The participant must order the explanations based on how useful it is to understand the reasoning behind the model's decision (ties were allowed between explanations). This process was repeated with a total of eight reviews. Tables 7 and 8 show the final average rankings of the explainability techniques.

Table 7: Personal utility tests. Average ranking (±standard deviation) for each explainability technique.

Technique	Position
Terms frequency	1.6 ±0.4
SHAP	2.1±0.5
GPT-3	1.7±0.5

As can be seen in Table 7, the explanations based on Term frequency (1.6) and GPT-3 (1.7) are in very close positions, both being above the third method SHAP (2.1). It is possible that the preference of respondents for the first two methods is due to the fact that the format of their explanations is more accessible and descriptive for a larger part of the population. Grouping the results by areas of knowledge (Table 8), we can see that the general trend is maintained for most areas. We can highlight the case of Social and Legal Sciences and Health Sciences, areas in which the positions in the ranking of Terms frequency and GPT-3 techniques are slightly inverted, with GPT-3 being the preferred option.

Table 8: Personal utility tests. Average ranking (± standard deviation) for each technique and area of knowledge.

Area of knowledge	Technique	Position
Engineering and Architecture	Terms frequency	1.6 ±0.4
	SHAP	2.2 ±0.5
	GPT-3	1.7 ±0.5
Social and Legal Sciences	Terms frequency	1.7±0.4
	SHAP	1.9±0.4
	GPT-3	1.6±0.5
Natural Sciences	Terms frequency	1.6±0.4
	SHAP	2.0±0.5
	GPT-3	1.7±0.5
Arts and Humanities	Terms frequency	1.5±0.4
	SHAP	1.9±0.5
	GPT-3	1.8±0.5
Health Sciences	Terms frequency	1.8±0.3
	SHAP	2.1±0.5
	GPT-3	1.6±0.5

5. Conclusion

In this work, we have proposed a pipeline for detecting anomalous reviews associated with Amazon products, which can be directly extrapolated to other online review platforms or scenarios with similar characteristics. The representation of the reviews using MPNet embeddings has enabled the training of classical anomaly detection algorithms that have achieved a very good performance. These have been evaluated using reviews from different products and categories, and the score they emit allows us to sort the reviews based on their normality.

A technique based on the occurrence of frequent terms has been proposed to generate explanations associated with the classifications of the reviews. This technique has been compared with SHAP, one of the reference post-hoc techniques in the field of explainability, and with GPT-3, due to its high power and versatility. To evaluate this aspect of the pipeline, we conducted a two-part survey in which 241 members of the university community participated.

From the first part of the explainability test we can conclude that, in general terms, the effect of the explanations has not been beneficial for the users. In any case, these tests allow us to reflect on the difficulty of using explainability and evaluation techniques in borderline scenarios where subjectivity plays an important role, such as the one presented in this article or in other fields of NLP, as well as in areas such as image or audio generation.

Regarding the second part of the explainability test, we have been able to conclude that respondents preferred explanations that presented a more natural and familiar appearance over more condensed and concise explanations such as those provided by SHAP, regardless of the explanation effect they provide. Explanations based on term frequency analysis have been preferred by respondents along with GPT-3, however, our approach presents a significantly lower computational costs and both its use and the explanations produced are simpler for the users.

As future work, it would be interesting to evaluate GPT-3 or other large models carrying out the complete process followed by the pipeline proposed in this work, instead of being tested only in the explainability module. We have not carried out this test due to the high computational cost that would be involved in processing the thousands of reviews to be evaluated using GPT-3. Another interesting possible line of future work would be to broaden the scope of the survey, both in terms of the number of products involved and the number of reviews, in order to clarify the conclusions reached at the forward simulation stage. Lastly, it would be very useful to try presenting the explanations issued by SHAP in a more familiar or natural format for the end user, so that we can see if their level of preference is increased for the general public.

Acknowledgements

This work was supported in part by grant *Machine Learning on the Edge - Ayudas Fundación BBVA a Equipos de Investigación Científica 2019* ; the Spanish National Plan for Scientific and Technical Research and Innovation (PID2019-109238GB-C22 and TED2021-130599A-I00); the Xunta de Galicia (ED431C 2022/44) and ERDF funds. CITIC, as a Research Center of the University System of Galicia, is funded by Consellería de Educación, Universidade e Formación Profesional of the Xunta de Galicia, Spain through the European Regional Development Fund (ERDF) and the Secretaría Xeral de Universidades (Ref. ED431G 2019/01).

References

[1] B. von Helversen, K. Abramczuk, W. Kopeć, R. Nielek, Influence of consumer reviews on online purchasing decisions in older and younger adults, Decision Support Systems 113 (2018) 1–10. doi:https://doi.org/10.1016/j.dss.2018.05.006.

[2] Amazon targets fake review fraudsters on social media, https://www.aboutamazon.com/news/policy-news-views/amazon-targets-fake-review-fraudsters-on-social-media.

[3] Tripadvisor review transparency report 2021, https://www.tripadvisor.com/TransparencyReport2021.

[4] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (jul 2009). doi:10.1145/1541880.1541882.

[5] A. Chernyavskiy, D. Ilvovsky, P. Nakov, Transformers:"the end of history" for natural language processing?, in: Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21, Springer, 2021, pp. 677–693.

[6] S. Tabinda Kokab, S. Asghar, S. Naz, Transformer-based deep learning models for the sentiment analysis of social media data, Array 14 (2022) 100157. doi:10.1016/j.array.2022.100157.

[7] Y. Kim, S. Bang, J. Sohn, H. Kim, Question answering method for infrastructure damage information retrieval from textual data using bidirectional encoder representations from transformers, Automation in Construction 134 (2022) 104061. doi:10.1016/j.autcon.2021.104061.

[8] Amazon customer reviews dataset, https://nijianmo.github.io/amazon/index.html.

[9] P. Schneider, F. Xhafa, Chapter 9 - anomaly detection, classification and cep with ml methods: Machine learning pipeline for medicine, in: P. Schneider, F. Xhafa (Eds.), Anomaly Detection and Complex Event Processing over IoT Data Streams, Academic Press, 2022, pp. 193–233. doi:10.1016/B978-0-12-823818-9.00020-1.

[10] H. T. Truong, B. P. Ta, Q. A. Le, D. M. Nguyen, C. T. Le, H. X. Nguyen, H. T. Do, H. T. Nguyen, K. P. Tran, Light-weight federated learning-based anomaly detection for time-series data in industrial control systems, Computers in Industry 140 (2022) 103692. doi:10.1016/j.compind.2022.103692.

[11] W. Hilal, S. A. Gadsden, J. Yawney, Financial fraud: A review of anomaly detection techniques and recent advances, Expert Systems with Applications 193 (2022) 116429. doi:10.1016/j.eswa.2021.116429.

[12] T. T. Huong, T. P. Bac, D. M. Long, T. D. Luong, N. M. Dan, L. A. Quang, L. T. Cong, B. D. Thang, K. P. Tran, Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach, Computers in Industry 132 (2021) 103509. doi:10.1016/j.compind.2021.103509.

[13] R. Mohawesh, S. Xu, S. N. Tran, R. Ollington, M. Springer, Y. Jararweh, S. Maqsood, Fake reviews detection: A survey, IEEE Access 9 (2021) 65771–65802. doi:10.1109/ACCESS.2021.3075573.

[14] N. Jindal, B. Liu, Opinion spam and analysis, in: Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM '08, Association for Computing Machinery, New York, NY, USA, 2008, p. 219–230. doi:10.1145/1341531.1341560.

[15] J. Salminen, C. Kandpal, A. M. Kamel, S. gyo Jung, B. J. Jansen, Creating and detecting fake reviews of online products, Journal of Retailing and Consumer Services 64 (2022) 102771. doi:10.1016/j.jretconser.2021.102771.

[16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (8) (2019) 9.

[17] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). arXiv:1907.11692.

[18] Şule Öztürk Birim, I. Kazancoglu, S. Kumar Mangla, A. Kahraman, S. Kumar, Y. Kazancoglu, Detecting fake reviews through topic mod-

elling, Journal of Business Research 149 (2022) 884–900. doi:10.1016/j.jbusres.2022.05.081.

[19] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.

[20] D. Vidanagama, A. Silva, A. Karunananda, Ontology based sentiment analysis for fake review detection, Expert Systems with Applications 206 (2022) 117869.

[21] L. Ruff, Y. Zemlyanskiy, R. Vandermeulen, T. Schnake, M. Kloft, Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4061–4071. doi:10.18653/v1/P19-1398.

[22] J. Mu, X. Zhang, Y. Li, J. Guo, Deep neural network for text anomaly detection in siot, Computer Communications 178 (2021) 286–296. doi:10.1016/j.comcom.2021.08.016.

[23] B. Song, Y. Suh, Narrative texts-based anomaly detection using accident report documents: The case of chemical process safety, Journal of Loss Prevention in the Process Industries 57 (2019) 47–54. doi:10.1016/j.jlp.2018.08.010.

[24] S. Seo, D. Seo, M. Jang, J. Jeong, P. Kang, Unusual customer response identification and visualization based on text mining and anomaly detection, Expert Systems with Applications 144 (2020) 113111. doi:10.1016/j.eswa.2019.113111.

[25] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[26] D. Novoa-Paradela, O. Fontenla-Romero, B. Guijarro-Berdiñas, Fast deep autoencoder for federated learning, Pattern Recognition 143 (2023) 109805. doi:https://doi.org/10.1016/j.patcog.2023.109805. URL https://www.sciencedirect.com/science/article/pii/S0031320323005034

[27] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space (2013). doi:10.48550/ARXIV.1301.3781.

[28] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543. doi:10.3115/v1/D14-1162.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[30] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding (2018). doi:10.48550/ARXIV.1810.04805.

[31] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Advances in neural information processing systems 32 (2019).

[32] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[33] OpenAI, GPT-4 technical report (2023). arXiv:2303.08774.

[34] Hugging face, https://huggingface.co/, accessed: 2023-05-09.

[35] S. S. Khan, M. G. Madden, One-class classification: taxonomy of study and review of techniques, The Knowledge Engineering Review 29 (3) (2014) 345–374.

[36] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, L. Bottou, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion., Journal of machine learning research 11 (12) (2010).

[37] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115. doi:10.1016/j.inffus.2019.12.012.

[38] N.-y. Liang, G.-b. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Transactions on Neural Networks 17 (6) (2006) 1411–1423. doi:10.1109/TNN.2006.880583.

[39] Y. Wang, J. Wong, A. Miner, Anomaly intrusion detection using one class svm, in: Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004., 2004, pp. 358–364. doi:10.1109/IAW.2004.1437839.

[40] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017.
URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[41] P. Hase, M. Bansal, Evaluating explainable AI: Which algorithmic explanations help users predict model behavior?, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5540–5552. doi:10.18653/v1/2020.acl-main.491.

Appendix A. Hyperparameters used during training

This appendix contains the values of the hyperparameters finally chosen as the best for each method and dataset, listed in Tables A.9 and A.10.

DAEF [26], OS-ELM [38], and OC-SVM [39] respectively.

- Deep Autoencoder for Federated learning (DAEF)[26].
  - Architecture: Neurons per layer.

- λ_{hid}: Regularization hyperparameter of the hidden layer.
- λ_{last}: Regularization hyperparameter of the last layer.
- μ: Anomaly threshold.

• Online Sequential Extreme Learning Machine (OS-ELM)[38].

- Architecture: Neurons per layer.
- μ: Anomaly threshold.

• One-Class Support Vector Machine (OC-SVM)[39].

- An upper bound on the fraction of training errors and a lower bound of the fraction of support vectors (ν).
- Kernel type: Linear, Polynomial or RBF.
- Kernel coefficient γ (in the case of polynomial and RBF kernels).
- Degree (in the case of polynomial kernel).

Normal class	DAEF	OS-ELM	OC-SVM
Chocolate bars	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: outlier IQR	Arch: [768, 400, 768], μ: extreme IQR	ν: 0.1, kernel: rbf, γ: 0.00097
Colored pencils	Arch: [768, 550, 650, 768], λ_{hid}: 0.1, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 500, 768] , μ: extreme IQR	ν: 0.1, kernel: poly, degree: 2, γ: scale
Gaming mouse	Arch: [768, 550, 650, 768], λ_{hid}: 0.1, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 100, 768], μ: outlier IQR	ν: 0.1, kernel: rbf, γ: scale
Bluetooth speaker	Arch: [768, 550, 650, 768], λ_{hid}: 0.75, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 20, 768], μ: outlier IQR	ν: 0.1, kernel: sigmoid, γ: scale
Foot insoles	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: outlier IQR	Arch: [768, 300, 768]], μ: outlier IQR	ν: 0.1, kernel: rbf, γ: scale

Table A.9: Hyperparameters used during the 1vs.4 experimentation.

Normal class	DAEF	OS-ELM	OC-SVM
Chocolate bars	Arch: [768, 550, 650, 768], λ_{hid}: 0.1, λ_{last}: 0.1, μ: Q_{90}	Arch: [768, 400, 768], μ: outlier IQR	ν: 0.1, kernel: poly, degree: 3, γ: scale
Anise seeds	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{80}	Arch: [768, 500, 768] , μ: outlier IQR	ν: 0.1, kernel: poly, degree: 4, γ: scale
Colored pencils	Arch: [768, 550, 650, 768], λ_{hid}: 0.25, λ_{last}: 0.25, μ: outlier IQR	Arch: [768, 500, 768], μ: extreme IQR	ν: 0.1, kernel: rbf, γ: scale
Ergonomic cushion	Arch: [768, 550, 650, 768], λ_{hid}: 0.5, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 500, 768], μ: extreme IQR	ν: 0.1, kernel: sigmoid, γ: scale
Gaming mouse	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{90}	Arch: [768, 500, 768]], μ: extreme IQR	ν: 0.1, kernel: poly, degree: 3, γ: scale
PS4 membership	Arch: [768, 550, 650, 768], λ_{hid}: 0.1, λ_{last}: 0.1, μ: Q_{90}	Arch: [768, 400, 768], μ: extreme IQR	ν: 0.1, kernel: rbf, γ: 0.00097
Bluetooth speaker	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{90}	Arch: [768, 500, 768] , μ: Q_{90}	ν: 0.2, kernel: linear, γ: scale
Wi-Fi range extender	Arch: [768, 550, 650, 768], λ_{hid}: 0.25, λ_{last}: 0.1, μ: Q_{90}	Arch: [768, 500, 768], μ: outlier IQR	ν: 0.1, kernel: poly, degree: 4, γ: scale
Foot insoles	Arch: [768, 550, 650, 768], λ_{hid}: 0.75, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 200, 768], μ: outlier IQR	ν: 0.1, kernel: poly, degree: 4, γ: scale
Yoga leggings	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{90}	Arch: [768, 500, 768]], μ: extreme IQR	ν: 0.1, kernel: poly, degree: 4, γ: scale

Table A.10: Hyperparameters used during the 1vs.1 experimentation.