ColPos at HOMO-MEX 2024: Weighted Naive Bayes for LGBTQ+Phobia Detection in Spanish Text

Daniel Ayala Niño[1,*,†], Manuel Montes y Gómez[2,†] and Ciro Velasco Cruz[1,†]

[1]*Colegio de Postgraduados Campus Montecillo, Posgrado en Socieconomía, Estadística e Informática-Cómputo Aplicado, Carretera México-Texcoco km 36.5, Montecillo, Texcoco, Estado de México, México. C. P. 56230*

[2]*Department of Computational Sciences National Institute of Astrophysics, Optics and Electronics Puebla, México, 72840*

Abstract

This paper presents the research conducted by ColPos team on the HOMO-Mex 2024 shared task, focusing on the detection of LGBTQ+ phobic messages in online content. We leveraged a weighted Naive Bayes model, enhancing the traditional algorithm by incorporating weights for words and documents. Our work targeted both Track 1 (hate speech identification) and Track 3 (homophobic lyric detection) of the competition. In Track 1, our model achieved an F1-score of 0.791, placing us 11th out of 17 teams. For Track 3, we obtained an F1-score of 0.489, securing 7th place out of 13. Our results demonstrate the effectiveness of our weighted approach in outperforming the baseline Multinomial Naive Bayes model. This research emphasizes the critical need for timely identification of LGBTQ+phobic content to improve moderation efforts on social media and music platforms, fostering safer online environments.

Keywords

Naive Bayes, Text classification, LGBTQ+Phobia Detection, NLP

1. Introduction

The rise of hate speech worldwide, encompassing xenophobia, racism, antisemitism, anti-Muslim hatred, anti-LGBTQ+ hatred, misogyny, and other forms of intolerance, is a pressing issue that has been exacerbated by the rapid spread of social media [1]. These platforms, while revolutionizing global communication, have also become conduits for offensive and discriminatory language, including homophobic and transphobic comments. By analyzing user behavior on social media, researchers can gain insights into the prevalence and patterns of such harmful speech, aiding in the development of models to detect and mitigate homophobia effectively [2]. One way to do this is by applying Natural Language Processing (NLP) techniques, such as text classification.

In Mexico, the detection of homophobia has gained importance due to persistent violence and discrimination against the LGBTQ+ community [3][4]. To address this problem, initiatives like the IberLEF 2023 shared task: "HOMO-MEX: Hate speech detection in Online Messages directed towards the MEXican Spanish-speaking LGBTQ+ population" has been introduced [5]. This task comprised two tracks: one for classifying tweets as LGBT+phobic, not LGBT+phobic, or not LGBT+related (multi-class classification); and another for identifying specific types of LGBT+phobia, such as Lesbophobia, Gayphobia, Biphobia, Transphobia, or other LGBT+phobias (multi-label classification). These tracks leverage advancements in natural language processing and machine learning to detect and classify tweets containing LGBTQ+ phobic content, expressed either aggressively or subtly. This year, IberLEF's 2024 shared task: "HOMO-MEX 2024: Hate speech detection towards the Mexican Spanish-speaking LGBT+ population." [6][7], added a third task: the homophobic lyrics detection track. This binary task aims to predict if a phrase from song lyrics contains LGBT+phobic hate speech, classifying them as either LGBT+phobic or not LGBT+phobic (binary classification).

IberLEF 2024, September 2024, Valladolid, Spain

*Corresponding author.

†These authors contributed equally.

ayala.daniel@colpos.mx (D. A. Niño); mmontesg@inaoep.mx (M. M. y. Gómez); cvelasco@colpos.mx (C. V. Cruz)

0000-0002-1032-7037 (D. A. Niño)

For more than 50 years traditional text classification models dominated the field. These traditional methods include statistics-based models such as Naive Bayes (NB), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM). Compared to earlier rule-based methods, these models offered significant improvements in accuracy and stability [8].

In contrast, Deep Learning methods (DL) have revolutionized text classification by automatically providing semantically meaningful representations for NLP tasks without the need for manual rule and feature design [9][10]. However, their black-box nature makes it difficult to understand their decision-making processes, optimize them, and explain their performance differences across datasets [8][11]. Conversely, classic machine learning models like linear regression, decision trees, and bayesian models, are interpretable [12], allowing for clear insights into feature importance and decision-making processes, although they may offer lower predictive performance compared to DL.

NB is a popular model often used because of its computational efficiency and relatively good predictive performance [13]. In its standard form, the model predicts class labels by estimating prior and conditional probabilities. However, it can be improved through several modifications.

One such modification involves relaxing the main characteristic of the NB model, which is the assumption of conditional independence, so it is assumed that each attribute can have at most another attribute related [14][15], which alleviates the issue of conditional independence not being satisfied in text classification. Another way of improving it is by fine tuning the conditional probabilities through an iterative process; if a document is mistakenly classified in the iteration t, conditional probability would be updated by a step $-\delta$, and $+\delta$ otherwise [16]. Unlike attribute selection, which removes irrelevant or redundant words from the vocabulary, attribute weighting is more flexible. It assigns a weight ranging between 0 and 1 to each word. In this method, prior and conditional probabilities are calculated as usual, but the weight of each word is incorporated into the NB classification stage. This enhances the model's ability to distinguish between classes based on the importance of different words [17][18]. Instance weighting assigns different weights to different documents, in this approach the prior and conditional probabilities are estimated including the weights of the documents, respectively. This method allows the model to account for the varying importance of different documents [19].

This paper presents the results obtained in the above mentioned shared task, using a modified NB model that combines the attribute and instance weighting approaches proposed by Zhang et. al. [20] for the first track (hate speech detection) and the third track (homophobic lyrics detection). The paper is organized as follows: Section 2 will provide a description of the methods applied for data preprocessing and how the training was performed. In Section 3, we present our results and evaluate some insights of our model, and in Section 4, we conclude the research.

2. Methodology

This section presents an overview of the three tasks comprising the HOMO-MEX 2024 shared task [6][[7]. We then describe the datasets used in each task, followed by details on the text preprocessing steps and the weighted Naive Bayes classifier [20] employed and the baseline Multinomial NB, for Track 1 and 3, and the metrics used to evaluate their performance.

2.1. Tasks and dataset description

Iberlef 2024 HOMO-MEX 2024 shared task, presented three tasks aimed at advancing the detection and classification of hate speech, particularly focused on LGBT+phobia in various textual formats. The tasks are described as follows:

- Track 1: Hate Speech Detection (Multi-class)
  - Task: Classifies individual tweets into three categories: LGBT+phobic (P), not LGBT+phobic but mentioning the community (NP), and not LGBT+ related (NR).
  - Evaluation: Based on the accuracy of assigning a single label to each tweet.

- Track 2: Fine-grained Hate Speech Detection (Multi-label)
  - Task: Identifies one or more specific types of LGBT+phobia (lesbophobia (L), gayphobia (G), biphobia (B), transphobia (T), other (O)) present in tweets containing hate speech.
  - Evaluation: Assesses the ability to assign multiple labels accurately to each tweet.
- Track 3: Homophobic Lyrics Detection (Binary)
  - Task: Determines whether a phrase from a song lyric contains LGBT+phobic hate speech (P) or not (NP).
  - Evaluation: Measures the accuracy of binary classification on song lyrics

We participated in the first and third tasks of the shared task. Each task consisted of three distinct phases:

- Development Phase: Participants were provided with approximately 80
- Training Phase: The complete training dataset was released, allowing participants to refine their models.
- Testing Phase: Unlabeled test data was provided for prediction. The organizers evaluated the submitted predictions on this unseen data.

The first task consisted of data from 11,000 public tweets written in Spanish in Mexico. The dates of these tweets downloaded are 01-01-2012 to 01-10-2022 [21]. The distribution of the labels for the first track is presented in the Table 1.

Table 1
Distribution of Tweet Labels per Phase

Phase	P	NP	NR
Development	862	4360	1778
Training	5482	2246	1072
Testing	?	?	?

An example of each label is shown in Table 2:

Table 2
Task 1: Tweets Examples of Phobic, Non-Phobic and Non-Related Content

Class	Tweet
Phobic (P)	Si festejar un gol es provocar, ya ponte faldita maricon #FueraAxelDelUni . No tiene ni idea de lo que escribe url
No Phobic (NP)	Techno Travesti es una pinche rolota!!
Not Related (NR)	@krlangaas Ola. Ez ke bi ke tienez unas piernaz muy bonitaz y me gustan demaciado.

For the third track, the dataset is composed of 1240 lyrics from Spanish songs extracted between 2015 and 2023. The distribution of the labels for this track is presented in 3.

Table 3
Task 3 Distribution of Tweet Labels per Phase

Phase	P	NP	Total
Development	40	560	600
Training	39	945	984
Testing	?	?	246

An example of each label is shown in 4

Table 4
Task 3: Fragment Lyric Examples of Phobic and Non-Phobic Content

Class	Twitt
Phobic (P)	...Aguas por que ya viene Dar Gaver Está buscando culos...
No Phobic (NP)	...Eran dulces tus labios Y húmedos tus ojos que decían...

2.2. Data preprocessing

Text preprocessing is not only an essential step to prepare the corpus for modeling but also a key area that directly affects the natural language processing (NLP) application results [22].

The same preprocessing methods were applied to both Track 1 and Track 3. All documents (tweets from Track 1 and lyrics from Track 3) were converted to lowercase. Subsequently, we eliminated tokens containing hashtags (#) and tags (@) to focus on the core textual content. Additionally, words with special characters or numeric characters were discarded, along with any emojis.

After cleaning the data, it is divided into training and testing sets as provided. Hapax words (words appearing only once in the corpus) are removed from the vocabulary set, and any documents that become empty due to this removal are also eliminated from the training set. This is done because hapax words typically have limited statistical value for modeling [23].

When training a Naive Bayes (NB) model, it is limited to the data encountered during the training phase. This limitation leads to zero counts for Out of Vocabulary (OOV) words (words not present in the training set) which can reduce the classifier's accuracy [24][25]. This problem of zero counts for OOV words is particularly pronounced in the case of Twitter, which is rich in slang. In contrast to DL pre-trained models that have shown excellent performance in handling OOV cases [26].

To deal with this problem we use a fasText pre-trained model [27]. The fasText model was trained on 157 languages, including Spanish. For each OOV word in the test set, we find its word vector representation from the fasText model and replace it with the most similar word from the training vocabulary. This similarity is determined using cosine similarity, that ranges in the interval $[-1, 1]$. It is a measure that calculates the cosine of the angle between two word vectors, with higher values indicating greater similarity.

2.3. Multinomial Naive Bayes

Multinomial NB is a widely applied algorithm for classification with great effectiveness. It is assumed in NB that all features (from now on called words) of a document are independent given the class. In order to classify a new test document $\mathbf{x}$ denoted as a vector of attributes $< a_1, a_2, \ldots, a_m >$. Multinomial NB utilizes the highest posterior probability, equation 1, to predict its class label.

$$c(\mathbf{x}) = \arg\max_{c \in C} \log P(c) + \sum_{j=1}^{m} \log P(a_j|c) \tag{1}$$

Where C is the collection of all possible class labels c, m is the number of words in our vocabulary, a_j is the value for the jth A_j (which have two cases $a_j = 0$ absence of the word, and $a_j > 0$ its frequency otherwise) word of $\mathbf{x}$, $P(c)$ is the prior probability of the class c, and $P(a_j|c)$ is the conditional probability of $A_j = a_j$ given the class c, which is obtained by maximum likelihood estimation in equations 2 and 3, which includes Laplacian smoothing.

$$P(c) = \frac{\sum_{i=1}^{n} \delta(c_i, c) + \frac{1}{q}}{n + 1}, \tag{2}$$

$$P(a_j|c) = \frac{\sum_{i=1}^{n} \delta(a_{ij}, a_j) a_{ij} \delta(c_i, c) + \frac{1}{n_j}}{\sum_{i=1}^{n} \delta(c_i, c) + 1} \tag{3}$$

Where n is the number of training documents, q is the number of classes, c_i is the class label of the ith training instance, a_{ij} is the jth word frequency of the ith training instance, n_j is the number of states for the jth attribute A_j , and $\delta()$ is a binary function, which is 1 if its two parameters are identical and 0 otherwise, i.e. if word a_j is in document i, then $\delta(a_i, a_{ij}) = 1$.

NB makes the conditional independence assumption that all attributes are fully independent given the class. Since the assumption required by NB hardly holds true in real-world applications, Zhang et. al. [20] proposed a weighted NB to relax this assumption, they proposed a weight for the words and for each document in a way to relax the independence assumption. This method will be addressed in the next section.

2.4. Weighted Multinomial Naive Bayes

2.4.1. Words weights

Word weights (w_j^{att}) are directly incorporated into the Multinomial NB classification formula. This means that the contribution of each word to the classification decision is scaled according to its weight, equation 4. These weights will be in the $[0, 1]$ range. Words with lower weight values have a greater influence on the predicted class label, while attributes with higher weights have a lesser influence. Here, the mutual information is used to measure the normalized value reflecting the attribute-class relevance and the average attribute-attribute redundancy.

$$c(\mathbf{x}) = \arg\max_{c \in C} \log P(c) + \sum_{j=1}^{m} w_j^{att} \log P(a_j|c) \tag{4}$$

The process of word weighting can be summarized as follows:

1. **Calculate Mutual Information:** Mutual information (MI) is used to measure the correlation between each pair of random discrete variables. The word-class relevance and the word-word inter-correlation are respectively defined as:

$$I(A_j; C) = \sum_{a_j} \sum_{c} P(a_j, c) \log \frac{P(a_j, c)}{P(a_j)P(c)} \tag{5}$$

$$I(A_j; A_k) = \sum_{a_j} \sum_{a_k} P(a_j, a_k) \log \frac{P(a_j, a_k)}{P(a_j)P(a_k)} \tag{6}$$

2. **Normalize Mutual Information:** The calculated MI values are normalized to ensure they are comparable across different attributes.

$$I(A_j; C) = \sum_{a_j} \sum_{c} P(a_j, c) \log \frac{P(a_j, c)}{P(a_j)P(c)} \tag{7}$$

$$I(A_j; A_k) = \sum_{a_j} \sum_{a_k} P(a_j, a_k) \log \frac{P(a_j, a_k)}{P(a_j)P(a_k)} \tag{8}$$

3. **Calculate Word Weight:** The weight of each word is defined as the difference between its normalized word-class relevance and its average normalized word-word redundancy. This captures how relevant a word is to the class while accounting for redundancy with other words.

$$\Delta w_j^{att} = \underbrace{NI(A_j; C)}_{\text{relevance}} - \underbrace{\frac{1}{m-1} \sum_{\substack{k=1 \\ k \neq j}}^{m} NI(A_j; A_k)}_{\text{average redundancy}} \tag{9}$$

4. **Apply Sigmoid Transformation:** The calculated word weights may be negative. To ensure they fall within the range of 0 to 1, a standard logistic sigmoid function is applied.

The resulting weights indicate the relative importance of each word for the classification task.

2.5. Documents weights

In order to preserve the computational efficiency of Naive Bayes, a simplified approach is adopted to assign weights to documents. This method leverages the frequency of words within documents and the overall dataset to calculate document weights. The underlying idea is that the weight of a training document is positively correlated with the frequency of its word and the total number of distinct word frequencies in the entire training dataset.

At first, we focus on the frequency of each word, and then employ equation 10:

$$f_{ij} = \frac{\sum_{r=1}^{n} \delta(a_{rj}, a_{ij}) a_{rj}}{n} \tag{10}$$

where f_{ij} is the average frequency of the word aj across all documents

Then, let n_j be the number of states of the jth word, and the word state vector can be represented by $< n1, n2, \ldots, n_m >$. Then, the weight of the ith training document is defined as the inner product (scalar product) of its word value vector and its word frequency number vector.

$$w_i^{ins} = \sum_{j=1}^{m} (f_{ij} \times n_j) \tag{11}$$

The document weights would be included during the training:

$$P(c) = \frac{\sum_{i=1}^{n} \delta(c_i, c) w_i^{ins} + \frac{1}{q}}{n+1}, \tag{12}$$

$$P(a_j|c) = \frac{\sum_{i=1}^{n} \delta(a_{ij}, a_j) a_{ij} w_i^{ins} \delta(c_i, c) + \frac{1}{n_j}}{\sum_{i=1}^{n} \delta(c_i, c) + 1} \tag{13}$$

3. Results and discussions

3.1. Data preprocessing

As described in the Methodology section, the same preprocessing steps were applied to the data from both Track 1 and Track 3. The vocabulary size and final dataset size is described in Table 5. As we can observe in Table 5, a subset of vectors in the Track 1 training set resulted in zero vectors due to the exclusive presence of hashtags, user tags, or hapax.

Table 5
Track 1 and 3 Train set distribution after data preprocessing

Track	Number of documents	Vocabulary size
1	8798	8006
3	984	13505

Next we show an example of the preprocessed data for both Tracks.

- **Track 1**:
  - Before preprocessing: "#CuandoMiMamaDice Quien es ese gay que este al aire contigo @ChenchoRios"
  - After preprocessing: "quien es ese gay que este al aire contigo"
- **Track 3**:
  - Before preprocessing: "[Intro: Babo] Volvió el Don Vergas Bien relaja'o, enjaraba'o, njaraba'o Mora'o…"

– After prerpocessing: "intro babo volvió el don vergas bien relajao enjarabao enjarabao morao…"

For the OOV words we use the pre-trained fastText [27] vector representations of size 300. Each word in our vocabulary has its own vector, allowing us to compare them with OOV words using cosine similarity. OOV words were then replaced with vocabulary words having a cosine similarity higher than 0.6. An illustrative example from Track 1 is shown in Table 6.

Table 6
OOV Replacement Example

Version	Tweet
Original	url Adolescencias robadas, oprimidas, violentadas.
Preprocessed	adolescencias robadas oprimidas violentadas
OOV Replacement	infancias robando trabajadoras asesinadas

3.2. Experimental results

We were provided with labeled training data and unlabeled test data for model training, as shown in Table 1 and Table 3. In the next subsection, we will compare the performance of the base model (Multinomial NB), our proposed model (Weighted MN), and the HOMO-MEX 2024 base model. Additionally, we will analyze the differences in what our proposed model learned compared to the standard Multinomial Naive Bayes.

3.2.1. Conditional probabilities and word weights

NB models allow us to easily inspect the conditional probabilities of individual words. By examining which words have higher conditional probabilities within a particular class, we can gain insights into the words that are most indicative of that class.

The Weighted NB [20] added a weight to the words and documents according to the difference among their relevance and redundancy. From equation we can infer that the close the weight is to 0 the close the conditional probability $p(a|c)$ will be to 1. Table 7 present the 5 words with the highest and lowest weights, along with their conditional probabilities in both the Weighted NB model and the baseline Multinomial NB model.

Table 7
Track 1 top 5 and bottom 5 words with their normalized conditional probabilities $p_W(a_j|c)$ (proposed method) and $p_b(a_j|c)$ (NB base model), 0= NR, 1=NP, 2=P

Word (w_j^{att})	$p_W(a_j\|c=0)$	$p_W(a_j\|c=1)$	$p_W(a_j\|c=2)$	$p_b(a_j\|c=0)$	$p_b(a_j\|c=1)$	$p_b(a_j\|c=2)$
puta (1)	0.871	0.026	0.103	0.888	0.035	0.077
gay (1)	0.006	0.545	0.450	0.013	0.667	0.320
mujeres (1)	0.084	0.650	0.266	0.091	0.694	0.215
puto (1)	0.343	0.038	0.619	0.475	0.053	0.472
joto (1)	0.000	0.099	0.901	0.003	0.157	0.840
empoperate (0.249)	0.219	0.263	0.517	0.265	0.266	0.464
ligarde (0.249)	0.218	0.265	0.515	0.265	0.269	0.464
páramo (0.249)	0.218	0.267	0.514	0.265	0.269	0.464
conocidos (0.249)	0.214	0.278	0.506	0.265	0.269	0.464
vergones (0.249)	0.214	0.280	0.505	0.265	0.266	0.465

Table 7 reveals that words with higher weights often share semantic relationships with other words, resulting in less informative class distributions. Conversely, words with lower weights tend to have

distinct distributions among words, making them more informative for classification. The same occurs for Track 3.

3.3. Document weights

The document weights aimed to enhance the estimation of the prior and conditional probabilities (equations 2 and 3) by providing more information to the estimators based on the origin of each word during the training phase (equations 12 and 13).

Table 7 presents a comparison of the estimated conditional probabilities for the Weighted NB and Multinomial NB models. As shown in the table, the conditional probabilities from our proposed model improved compared to the base model, particularly for words strongly associated with class P, like "puta", "gay", "puto", and "joto," and for the word "vergones." Equations 10 and 11 demonstrate that the document weight is proportional to the number of words in it, meaning that longer documents receive a higher weight. However, this weighting scheme may not be the most effective way to differentiate between documents, as it does not account for the specific content or relevance of the words.

3.4. Evaluation for prediction performance

The evaluation metrics used were macro-averaged F1-score, precision, and recall. The models were compared with the HOMO-MEX 2024 base model. Our base model (Multinomial NB) was evaluated using only the macro-averaged F1-score on the test set. The results are shown in Table 8 and 9.

Table 8
Comparison of Model Performance on Track 1

Model	F1-score	Precision	Recall
Base NB	0.735	-	-
Weighted NB	0.791	0.83	0.763
HOMO-MEX 2024	0.852	0.916	0.818

Table 9
Comparison of Model Performance on Task 3

Model	F1-score	Precision	Recall
Base NB	0.489	-	-
Weighted NB	0.489	0.479	0.5
HOMO-MEX 2024	0.489	0.479	0.5

4. Conclusions

In this paper, we presented our approach for the HOMO-MEX 2024 shared task, focusing on the detection of LGBTQ+phobic content in Spanish text. Our weighted Naive Bayes model, incorporating both word and document weights, demonstrated its effectiveness in outperforming the baseline Multinomial Naive Bayes model on Track 1 (hate speech identification).

Our results highlight the potential of weighted Naive Bayes as a valuable tool for identifying and mitigating LGBTQ+phobic content online. The model's ability to assign weights to both words and documents allows it to capture nuanced patterns in language and better differentiate between hateful and non-hateful content.

While our model showed promising results, there is still room for improvement. Future work could explore alternative weighting schemes, incorporate additional features, or experiment with

different machine learning algorithms to further enhance the accuracy and robustness of LGBTQ+phobia detection systems.

This research contributes to the ongoing efforts to create safer and more inclusive online environments by providing a practical and effective approach for identifying and addressing harmful content. By continuing to develop and refine these tools, we can work towards a future where online platforms are free from hate speech and discrimination.

References

[1] United Nations Educational, Scientific and Cultural Organization, Addressing hate speech through education: a guide for policy-makers, 2024. URL: https://unesdoc.unesco.org/ark:/48223/pf0000384872, accessed: 2024-05-27.

[2] A. Hürriyetoğlu, H. Tanev, V. Zavarella, J. Piskorski, R. Yeniterzi, O. Mutlu, D. Yuret, A. Villavicencio, Challenges and applications of automated extraction of socio-political events from text (CASE 2021): Workshop and shared task report, in: A. Hürriyetoğlu (Ed.), Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021), Association for Computational Linguistics, Online, 2021, pp. 1–9. URL: https://aclanthology.org/2021.case-1.1. doi:10.18653/v1/2021.case-1.1.

[3] L. Ortiz-Hernández, J. Granados-Cosme, Violence against bisexuals, gays and lesbians in mexico city, J Homosex 50 (2006) 113–40. doi:10.1300/J082v50n04_06.

[4] I. Lozano-Verduzco, J. A. Fernandez-Nino, R. Baruch-Dominguez, Association between internalized homophobia and mental health indicators in lgbt individuals in mexico city, Salud Mental 40 (2017) 219–226. URL: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0185-33252017000500219&lng=es&nrm=iso. doi:10.17711/sm.0185-3325.2017.028.

[5] G. Bel-Enguix, H. Gómez-Adorno, G. Sierra, J. Vásquez, S. T. Andersen, S. Ojeda-Trueba, Overview of HOMO-MEX at Iberlef 2023: Hate speech detection in Online Messages directed toOwards the MEXican Spanish speaking LGBTQ+ population, Procesamiento del Lenguaje Natural 71 (2023).

[6] H. Gómez-Adorno, G. Bel-Enguix, H. Calvo, J. Vásquez, S. T. Andersen, S. Ojeda-Trueba, T. Alcántara, M. Soto, C. Macias, Overview of HOMO-MEX at Iberlef 2024: Hate Speech Detection Towards the Mexican Spanish Speaking LGBT+ Population, Procesamiento del Lenguaje Natural 73 (2024).

[7] L. Chiruzzo, S. M. Jiménez-Zafra, F. Rangel, Overview of IberLEF 2024: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024), co-located with the 40th Conference of the Spanish Society for Natural Language Processing (SEPLN 2024), CEUR-WS.org, 2024.

[8] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, L. He, A survey on text classification: From traditional to deep learning, ACM Trans. Intell. Syst. Technol. 13 (2022). URL: https://doi.org/10.1145/3495162. doi:10.1145/3495162.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.

[10] X. Bai, Text classification based on lstm and attention, in: 2018 Thirteenth International Conference on Digital Information Management (ICDIM), 2018, pp. 29–32. doi:10.1109/ICDIM.2018.8847061.

[11] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, 2019. arXiv:1811.10154.

[12] C. Molnar, G. Casalicchio, B. Bischl, Interpretable Machine Learning A Brief History, State of the Art and Challenges, Springer International Publishing, 2020, p. 417–431. URL: http://dx.doi.org/10.1007/978-3-030-65965-3_28. doi:10.1007/978-3-030-65965-3_28.

[13] J. Chen, H. Huang, S. Tian, Y. Qu, Feature selection for text classification with naïve bayes, Expert Systems with Applications 36 (2009) 5432–5435. URL: https://www.sciencedirect.com/science/article/pii/S0957417408003564. doi:https://doi.org/10.1016/j.eswa.2008.06.054.

[14] J. Wu, S. Pan, X. Zhu, P. Zhang, C. Zhang, Sode: Self-adaptive one-dependence estimators for classification, Pattern Recognition 51 (2016) 358–377. URL: https://www.sciencedirect.com/science/article/pii/S0031320315003118. doi:https://doi.org/10.1016/j.patcog.2015.08.023.

[15] D. Shen, J. Wu, B. Cao, J.-T. Sun, Q. Yang, Z. Chen, Y. Li, Exploiting term relationship to boost text classification, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 1637–1640. URL: https://doi.org/10.1145/1645953.1646192. doi:10.1145/1645953.1646192.

[16] K. El Hindi, H. AlSalman, S. Qasem, S. Al Ahmadi, Building an ensemble of fine-tuned naive bayesian classifiers for text classification, Entropy 20 (2018). URL: https://www.mdpi.com/1099-4300/20/11/857. doi:10.3390/e20110857.

[17] L. Jiang, L. Zhang, C. Li, J. Wu, A correlation-based feature weighting filter for naive bayes, IEEE Transactions on Knowledge and Data Engineering 31 (2019) 201–213. doi:10.1109/TKDE.2018.2836440.

[18] C.-H. Lee, An information-theoretic filter approach for value weighted classification learning in naive bayes, Data & Knowledge Engineering 113 (2018) 116–128. URL: https://www.sciencedirect.com/science/article/pii/S0169023X16301276. doi:https://doi.org/10.1016/j.datak.2017.11.002.

[19] L. JIANG, D. WANG, Z. CAI, Discriminatively weighted naive bayes and its application in text classification, International Journal on Artificial Intelligence Tools 21 (2012) 1250007. URL: https://doi.org/10.1142/S0218213011004770. doi:10.1142/S0218213011004770. arXiv:https://doi.org/10.1142/S0218213011004770.

[20] H. Zhang, L. Jiang, L. Yu, Attribute and instance weighted naive bayes, Pattern Recognition 111 (2021) 107674. URL: https://www.sciencedirect.com/science/article/pii/S0031320320304775. doi:https://doi.org/10.1016/j.patcog.2020.107674.

[21] J. Vásquez, S. Andersen, G. Bel-enguix, H. Gómez-adorno, S.-l. Ojeda-trueba, HOMO-MEX: A Mexican Spanish annotated corpus for LGBT+phobia detection on Twitter, in: Y.-l. Chung, P. R\"ottger, D. Nozza, Z. Talat, A. Mostafazadeh Davani (Eds.), The 7th Workshop on Online Abuse and Harms (WOAH), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 202–214. URL: https://aclanthology.org/2023.woah-1.20. doi:10.18653/v1/2023.woah-1.20.

[22] C. P. Chai, Comparison of text preprocessing methods, Natural Language Engineering 29 (2023) 509–553. doi:10.1017/S1351324922000213.

[23] A. Lardilleux, Y. Lepage, Hapax legomena: Their contribution in number and efficiency to word alignment, in: Z. Vetulani, H. Uszkoreit (Eds.), Human Language Technology. Challenges of the Information Society, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 440–450.

[24] T. Boroș, D. Ștefănescu, R. Ion, Handling Two Difficult Challenges for Text-to-Speech Synthesis Systems: Out-of-Vocabulary Words and Prosody: A Case Study in Romanian, Springer New York, New York, NY, 2013, pp. 137–161. URL: https://doi.org/10.1007/978-1-4614-6934-6_7. doi:10.1007/978-1-4614-6934-6_7.

[25] J. Awwalu, A. Bakar, M. Yaakub, Hybrid n-gram model using naïve bayes for classification of political sentiments on twitter, Neural Computing and Applications 31 (2019) 9207–9220. URL: https://doi.org/10.1007/s00521-019-04248-z. doi:10.1007/s00521-019-04248-z.

[26] A. Benamar, C. Grouin, M. Bothua, A. Vilnat, Evaluating tokenizers impact on OOVs representation with transformers models, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, S. Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 4193–4204. URL: https://aclanthology.org/2022.lrec-1.445.

[27] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning word vectors for 157 languages, 2018. URL: https://arxiv.org/abs/1802.06893. arXiv:1802.06893.