

IU-NLP-JeDi: Investigating Sexism Detection in English and Spanish

Notebook for the EXIST Lab at CLEF 2023

Matthew Buzzell¹, Jeremy Dickinson¹, Natasha Singh¹ and Sandra Kübler¹

¹Indiana University, Bloomington, IN, USA

Abstract

In this paper we present the results from three different classification algorithms our team (IU-NLP-JeDi) developed for Task 1 of the EXIST 2023 shared task on Sexism Identification on Social Networks. The task consists of identifying sexism within English and Spanish tweets. We separated the English and Spanish tweets and then developed two different neural model approaches and an SVM model for each language. We achieved our highest ICM score on the test set from the RNN model.

Keywords

TweetTokenizer, SVM, RNN, CNN

1. Introduction

Sexism is defined as discrimination of a person or group based on sex. In many cases, these gender stereotypes assume a difference in social standing between men and women. However, this discrimination can be expressed explicitly (discrimination that is stated plainly) or implicitly (discrimination that is implied or obfuscated). The examples below from the EXIST 2023 dataset show the distinction between explicit and implicit sexism:

Explicit: Call me sexist all you want but no Nation ever succeeds with a woman as the Head. It's just the way it is. They final nail is already in the coffin.

Implicit: Wife material, wake up and cook for your husband.

Implicit sexism has been used in many online platforms to perpetuate gender stereotypes without risk of penalty from administrators of online platforms. In addition, the popularity and easy access of social media has only resulted in a further increase in content involving gender discrimination across the internet. Swim et al. [1] have shown that such prejudice impacts the performance of its victim in a tangibly negative way. Thus, detecting and limiting sexist behavior on online platforms has become a central topic of research in the field of computational linguistics. To contribute to the growing body of research on sexism detection, this paper participates in Task 1 of the sEXism Identification in Social neTworks (EXIST) 2023 shared task [2, 3] by training a binary classifier to predict if a given text has gender-bias. The

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece


✉ matabuzz@iu.edu (M. Buzzell); jedicki@iu.edu (J. Dickinson); singhnat@iu.edu (N. Singh); skuebler@indiana.edu (S. Kübler)

🌐 <https://cl.indiana.edu/~skuebler/> (S. Kübler)

🆔 0000-0003-0885-5436 (S. Kübler)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

goal of this shared task is to develop systems that can decide whether or not a given tweet contains sexist expressions or behaviors.

Our work focuses on investigating different machine learning models with regard to their suitability for the task. In this study, we conducted an in-depth analysis of the impact of several factors on the performance of various models. Specifically, we examined :

1. The effects of data pre-processing techniques on model performance.
2. The implications of utilizing word-level versus character-level models.
3. The use of soft labels versus hard labels during the training process.

These approaches were developed during a course on machine learning in NLP.

2. Related Work

There is a growing body of research on the topic of sexism detection. In line with our task, Vaca-Serrano [4] built a system for sexism identification and for sexism categorization for both English and Spanish. For the English tweets, DeBERTa-v3-large, RoBERTa-large and BERTweet-large were trained, while for Spanish tweets, BERTIN, MarIA-base, BETO, and RoBERTuito were trained. After training, Vaca-Serrano implemented ensemble learning using weighted majority voting over all the models to decide on the final classification for a tweet. For sexism identification, BERTweet-large reached the highest F1-score of 0.903 for English tweets while MarIA-base reached the highest F1-score of 0.883 for Spanish tweets. For sexism identification, DeBERTa-v3-large had the highest F1-score of 0.729 for English tweets while BETO had the highest F1-score of 0.820 for Spanish tweets. Another approach taken by Chiril et al. [5] examined the effectiveness of data augmentation methods based on sentence similarity and the use of gender stereotype detection for sexism classification on a multilingual data set. The best results were obtained by a SentenceBERT model trained to detect both sexism and gender stereotypes (multiclass classification), which achieved precision and recall scores of 0.816 and 0.827 respectively, outperforming a BERT model trained on word embeddings, linguistic features and generalization strategies. Differing from the previous two approaches, Jha and Mamidi [6] used an SVM and a sequence-to-sequence model to detect sexism according to ambivalent sexism theory [7], according to which sexism comes in hostile and benevolent forms. By training an SVM model and a sequence-to-sequence model on a dataset labeled for hostile and benevolent sexism they found that the SVM model outperformed the sequence-to-sequence model for benevolent tweets, while the sequence-to-sequence model outperformed the SVM model for the detection of hostile tweets.

3. Methodology

3.1. Data

We used the EXIST 2023 shared task training, validation, and test set provided by the shared task organizers. This dataset was constructed by compiling a list of over 400 commonly used sexist expressions in English and Spanish, and tweets containing these expressions were extracted for both languages. Each tweet was then classified by six crowd-sourced annotators as “sexist” or

“nonsexist”. Additionally, the annotators’ gender (male or female) and age group (18-22 years, 23-45 years, or 46 or more years) was recorded. Since these annotations were crowdsourced, the annotators were provided with guidelines created by two experts in gender issues. It is important to note that this dataset follows the learning with disagreements paradigm, i.e., there were no gold annotations as such provided. Given that six annotations were provided for each tweet, there were cases in which there was a tie for the majority class. To generate the hard labels for the dataset, any tweet labeled as sexist by three or more of the annotators was considered “sexist”.

3.2. Pre-Processing

HTML characters were converted to Unicode (e.g., the HTML “>” was converted to “>”, the text font was normalized to Roman characters, removing any non-Roman characters while ensuring Spanish characters with diacritics were not removed. Any emoticons were converted to their emoji equivalents and URL links were removed. Spaces were added between consecutive usernames, and any symbols were converted to their literals (e.g., the ° symbol to the word “degrees”). All special characters were removed before passing each tweet through NLTK’s TweetTokenizer. Additionally, any numbers or words with numbers were removed. Subsequently, all hashtags were passed through a parser that removed the “#” symbol and tokenized the contents of the hashtag using Wordninja (<https://github.com/keredson/wordninja>), a probabilistic parser of concatenated words that was trained on the Spanish Billion Words Corpus [8] and the Kaggle English Word Frequency dataset (<https://www.kaggle.com/datasets/rtatman/english-word-frequency>). Pairs of upside-down question/exclamation marks and rightside-up question/exclamation marks were condensed down to a single question mark and exclamation point respectively. Finally, duplicate usernames, exclamation points, and question marks were counted before being removed.

3.3. Models

The following classification algorithms were chosen for the task of sexism detection in Spanish and English: Support Vector Machines with term frequency-inverse document frequency (TF-IDF) using bigrams and trigrams. Recurrent Neural Network (RNN) with one embedding layer, two bidirectional long short-term memory layers, followed by a dense layer with softmax activation. A Convolutional Neural Network (CNN) model with an embedding layer, a convolutional layer, a max pooling layer, a flatten layer, and two dense layers with ReLU and sigmoid activations respectively.

3.4. Data Transformation & Feature Extraction

3.4.1. Support Vector Machines

Support Vector Machines seek to find a line or hyperplane that maximizes the margin between two classes that are projected into vector space [9]. The SVM models trained in the present study utilize a mixture of features for sexism detection. In this study, five SVM models were trained for Spanish, and five SVM models were trained for English, resulting in ten SVM models total.

Table 1
Optimal hyperparameters for each SVM model.

| Model | Lang. | C | Gamma | Kernel |
|--|-------|-----|-------|--------|
| raw SVM | EN | 1 | 1 | RBF |
| processed SVM | EN | 0.1 | 10 | Poly |
| raw SVM with upsampling | EN | 10 | 1 | RBF |
| processed SVM with upsampling | EN | 0.1 | 10 | Poly |
| processed SVM with upsampling & features | EN | 1 | 10 | Poly |
| raw SVM raw | ES | 1 | 0.1 | RBF |
| processed SVM | ES | 1 | 1 | Poly |
| raw SVM with upsampling | ES | 10 | 1 | RBF |
| processed SVM with upsampling | ES | 10 | 1 | RBF |
| processed SVM with upsampling & features | ES | 10 | 0.1 | Linear |

All models were trained using TF-IDF of word / character bigrams and trigrams. The additional features used to train the SVM models, except for the TF-IDF bigrams and trigrams, were obtained using the tokenizer described in section 3.2. This tokenizer counts and returns the number of usernames in each tweet, the number of exclamation points, questions marks, usernames used in the possessive, and the number of hashtags present in the tweet.

Furthermore, upsampling was conducted on two clean and two original datasets for each language. The upsampling step consists of duplicating the sexist tweets present in the training set. The purpose of conducting upsampling was to increase the number of sexist tweets to improve recall on the minority class (sexism).

To summarize, the following models were trained for each language, all with TF-IDF character bigrams and trigrams: one with clean data, one with the original data, one with upsampling on the clean data, and one with upsampling on the original data, and one with upsampling and the additional features described (username counts, hashtag counts, etc.) on the clean data.

For parameter optimization, we performed grid search to obtain the best regularization(C) and gamma parameter for each SVM model. The optimal parameters for each SVM model are shown in Table 1.

3.4.2. Neural Models

First, unique words from the tokenized input data are collected to form a vocabulary. Then two special tokens '[UNK]' and '[PAD]' are added to vocabulary. [UNK] is used to mask any word in test data which is not present in the vocabulary. [PAD] is used to make all the sentences of equal length. The vocabulary thus obtained is then used to encode the words in an input sentence to numbers based on the index at which they are present in the vocabulary. All unknown words are mapped to the index of [UNK] token. Finally, all the sentences are extended to a sentence of length 'max_len' (100 words or 300 chars) by adding [PAD] tokens at the end. Similarly, the output labels are one hot-encoded and are used to train the RNN and CNN model for 10 epochs.

Table 2

Official evaluation scores on test set.

| Model | Class. | Rank | Soft-soft | Hard-hard | F1 score | Hard-soft |
|---------------|--------|------|-----------|-----------|----------|-----------|
| IU-NLP-JeDi_3 | RNN | 31 | 0.1244 | 0.2753 | 0.6909 | -0.5071 |
| IU-NLP-JeDi_2 | CNN | 34 | -0.1499 | 0.1851 | 0.6485 | 0.4139 |
| IU-NLP-JeDi_1 | SVM | - | - | 0.2676 | 0.4839 | -0.5097 |

3.5. Evaluation Metrics

To allow for the comparison of the classification algorithms implemented in this study, the key metrics calculated were Accuracy, Precision, Recall and F1-score. The performance of the models were assessed primarily using the F1-score during the training and validation phase.

The final evaluation metric used to evaluate the performance of models on the test set was ICM metric [10]. ICM calculates the below scores for each model:

1. HARD-HARD: hard system output and hard ground truth.
2. HARD-SOFT: hard system output and soft ground truth.
3. SOFT-SOFT: soft system output and soft ground truth.

4. Results

4.1. Performance in the Official Evaluation

For the official ICM evaluation on the test set, we submitted the predictions obtained by an RNN model (IU-NLP-JeDi_3) that was trained on the word-level processed sequences with soft labels, a CNN model (IU-NLP-JeDi_2) that was trained on word-level raw sequence with soft labels, and an SVM model (IU-NLP-JeDi_1) that was trained using TF-IDF of character bigrams and trigrams of a processed sequence.

Table 2 shows the results of the official evaluation on the text set for these models. Among these models, the RNN model exhibits the highest performance across all three evaluation types, followed by the SVM model and then the CNN model. Notably, both the RNN model and the SVM model attained the highest ICM scores of 0.2753 and 0.2676, respectively, in the Hard-hard evaluation.

4.2. Performance on the Validation Set

We performed more extensive experiments on the validation set. Table 3 shows the results of the ICM evaluations for the models with the best performance. Table 4 shows the results of a range of CNN, RNN, and SVM experiments respectively. The best macro F1 scores for each model are marked in bold for each language.

Based on the macro F1-scores, the highest performing SVM model for English was trained on TF-IDF character bigrams and trigrams constructed from English tweets that had not been preprocessed. This English model included upsampling of the minority class and achieved a macro F1-score of 0.746. The best performing CNN and RNN models were trained using

Table 3

ICM evaluation scores on the validation set.

| Model | Level | Labels | Hard-hard | Hard-soft |
|-------------------------|-------|--------|-----------|-----------|
| processed RNN | word | soft | 0.2923 | 0.2831 |
| | word | hard | 0.1447 | -0.4564 |
| raw CNN | char | hard | -0.2652 | -0.9556 |
| | char | soft | -0.1883 | -0.8432 |
| | char | soft | -0.2491 | -0.9167 |
| | char | hard | -0.2687 | -1.0151 |
| | word | hard | 0.1853 | -0.0603 |
| | word | soft | 0.2104 | 0.0022 |
| processed SVM | char | hard | 0.2992 | -0.2773 |
| raw SVM with upsampling | char | hard | 0.2581 | -0.4038 |

Table 4

Macro F1 scores for the different models.

| Model | Lang. | Level | F1: CNN | F1: RNN | F1: SVM |
|-----------------------------------|-------|-------|--------------|--------------|--------------|
| raw | EN | word | 0.713 | 0.722 | 0.307 |
| | EN | char | 0.544 | 0.618 | 0.727 |
| raw + upsampling | EN | char | - | - | 0.746 |
| processed | EN | word | 0.721 | 0.755 | 0.379 |
| | EN | char | 0.556 | 0.556 | 0.739 |
| processed + upsampling | EN | char | - | - | 0.737 |
| processed + upsampling + features | EN | char | - | - | 0.686 |
| raw | ES | word | 0.654 | 0.701 | 0.392 |
| | ES | Char | 0.539 | 0.576 | 0.692 |
| raw + upsampling | ES | char | - | - | 0.707 |
| processed | ES | word | 0.702 | 0.685 | 0.372 |
| | ES | char | 0.601 | 0.641 | 0.740 |
| processed + upsampling | ES | char | - | - | 0.732 |
| processed + upsampling + features | ES | char | - | - | 0.701 |

word-level processed input sequences and hard-label outputs. These models achieved a macro F1-score of 0.721 and 0.7546 respectively. As for Spanish, the best performing SVM model was trained on TF-IDF character bigrams and trigrams taken from preprocessed Spanish tweets without any additional features. This model did not include upsampling and it reached a macro F1-score of 0.740. The highest performing CNN and RNN models were trained on word level raw sequences with hard labels. They attained an F1-score of 0.702 and 0.701 respectively.

For the ICM evaluation on the validation set, the prediction result of each model for both languages was consolidated and evaluated based on Hard-hard and Hard-soft scores. Among the various models, the SVM model trained using TF-IDF of processed input and the RNN model trained on word-level processed sequences and soft labels emerged as the top-performing models, exhibiting Hard-hard scores of 0.2992 and 0.2923, respectively. Only the RNN trained on word-level sequences and soft labels achieved a positive Hard-soft score of 0.2831. All the

remaining models performed poorly on this metric and obtained negative results.

4.3. Investigating the Effects of Preprocessing

Table 4 also shows the experiments using pre-processing for all models and for upsampling with the SVM. These results show that the cleaning and preprocessing step results in a higher performance for both languages. For English, the macro F1-score increases from 0.727 to 0.739, and for Spanish, it increases from 0.692 to 0.740. Upsampling sexist tweets also improves the SVM models. The results also show that preprocessing boosts the performance of the RNN and CNN models for English, but affects the performance negatively for Spanish. In the case of English, the RNN model's F1-score improves from 0.722 to 0.7546 and the CNN model's F1-score improves from 0.713 to 0.721. However, for Spanish, the RNN model's F1-score declines from 0.701 to 0.6849 and the CNN model's F1-score drops from 0.705 to 0.6947.

5. Limitation and Challenges

The main limitations in our study comes from two sources: (i) the inability of neural models, such as RNNs and CNNs, to handle long sequences and (ii) the information loss due to preprocessing. Some of the tweets in the EXIST 2023 dataset were around 100 words long and in such cases, the neural models are not able to effectively model long dependencies. Neural models tend to forget the initial words of the sequence, leading to a decline in model performance. This problem is more prominent when using a character level model where the sequence length was roughly 300 characters long. This was evident from the experiments as the character-level models consistently under-performed against the word-level models. The other major limitation of information loss due to preprocessing was a combined effect of non-standard spelling in the tweets, the presence of both English and Spanish words in the same tweet, and the presence of words and characters from other languages. All words which were not seen during the training phase were replaced by [UNK] token, thus preventing the models from accessing information which may be crucial in determining if a tweet was sexist or not. In our training setup, we created a separate model for English and Spanish tweets. Due to this, we categorized the tweets with both English and Spanish words as Spanish and treated English words as out of vocabulary words, thus replacing them with the token [UNK]. Lastly, some tweets had characters/words from other languages which were completely removed in our analysis. All these factors lead to an information loss for our models, hence decreasing the model's performance.

Additionally, there were many challenges we faced throughout our study, specifically focused on the creation of an accurate tokenizer. While analyzing the tweets, we encountered tweets using non-Roman characters, leading to issues when trying to tokenize these tweets. These included characters from other languages, such as Arabic, Gregorian, CJK, Hangul, and Hiranaga characters, and non-traditional styles of Roman characters, such as Gothic letters. To handle this issue, we used the unicode values of the characters we wanted and removed any characters outside of that range of unicode values. Another challenge with the tokenizer came from punctuation marks and the differences in punctuation between Spanish and English. In order to simplify processing, we decided to delete the Spanish inverted punctuation. However, punctuation tends to be irregular or missing in tweets. For this reason, we developed a heuristic

that deleted an inverted punctuation mark only if a regular one could be found in the tweet. We faced additional challenges when handling emoji, removing usernames and inconsistent diacritics. For all of those cases, we developed diacritics.

6. Conclusion and Future Work

In this study we demonstrated that when optimized on F1-score, the neural models trained on word level with hard labels and an SVM trained on TF-IDF of character bigrams and trigrams are our best performing models. However, when optimized on ICM metric, the neural models show better performance with soft labels and the behavior of the SVM remains unchanged. These findings illustrate the dependency of model behavior on the choice of evaluation metric. Additionally, we showed that applying upsampling techniques on the minority class can enhance the performance of SVM models when dealing with an imbalanced datasets.

For future work, we will investigate how to utilize the gender and age information of the annotators, either by modeling this latent variable in the models, or by choosing reliable training data or labels. Additionally, we will investigate whether the labels are influenced by the gender bias of the annotators.

References

- [1] J. K. Swim, L. L. Hyers, L. L. and Cohen, M. J. Ferguson, Everyday sexism: Evidence for its incidence, nature, and psychological impact from three daily diary studies, *Journal of Social Issues* 57 (2001) 31–53.
- [2] L. Plaza, J. Carrillo-de-Albornoz, R. Morante, E. Amigó, J. Gonzalo, D. Spina, P. Rosso, Overview of EXIST 2023 – Learning with Disagreement for Sexism Identification and Characterization, in: A. Arampatzis, E. Kanoulas, T. Tsirikika, S. Vrochidis, A. Giachanou, D. Li, M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Thessaloniki, Greece, 2023.
- [3] L. Plaza, J. Carrillo-de-Albornoz, R. Morante, E. Amigó, J. Gonzalo, D. Spina, P. Rosso, Overview of EXIST 2023 – Learning with Disagreement for Sexism Identification and Characterization (Extended Overview), in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), *Working Notes of CLEF 2023 – Conference and Labs of the Evaluation Forum*, 2023.
- [4] A. Vaca-Serrano, Detecting and classifying sexism by ensembling transformers models, in: *IberLEF@SEPLN*, 2022.
- [5] P. Chiril, F. Benamara, V. Moriceau, “be nice to your wife! the restaurants are closed”: Can gender stereotype detection improve sexism classification?, in: *Findings of the Association for Computational Linguistics: EMNLP*, 2021, pp. 2833–2844.
- [6] A. Jha, R. Mamidi, When does a compliment become sexist? Analysis and classification of ambivalent sexism using Twitter data, in: *Proceedings of the Second Workshop on NLP and Computational Social Science*, 2017, pp. 7–16.
- [7] P. Glick, S. T. Fiske, The ambivalent sexism inventory: Differentiating hostile and benevolent sexism, *Journal of Personality and Social Psychology* (1996) 491.

- [8] C. Cardellino, Spanish Billion Words Corpus and embeddings, 2019. <https://crscardellino.github.io/SBWCE/>.
- [9] N. Christianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, 2000.
- [10] E. Amigó, A. Delgado, Evaluating extreme hierarchical multi-label classification, in: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022, pp. 5809–5819.