Multi-label Categorization of Accounts of Sexism using a Neural Framework

Pulkit Parikh[1], **Harika Abburi**[1], **Pinkesh Badjatiya**[1], **Radhika Krishnan**[1], **Niyati Chhaya**[1,2], **Manish Gupta**[1*], **Vasudeva Varma**[1]
[1]IIIT-Hyderabad, India
[2]Adobe Research, India
{pulkit.parikh,harika.a,pinkesh.badjatiya}@research.iiit.ac.in,
radhika.krishnan@iiit.ac.in, nchhaya@adobe.com, {manish.gupta,vv}@iiit.ac.in

Abstract

Sexism, an injustice that subjects women and girls to enormous suffering, manifests in blatant as well as subtle ways. In the wake of growing documentation of experiences of sexism on the web, the automatic categorization of accounts of sexism has the potential to assist social scientists and policy makers in utilizing such data to study and counter sexism better. The existing work on sexism classification, which is different from sexism detection, has certain limitations in terms of the categories of sexism used and/or whether they can co-occur. To the best of our knowledge, this is the first work on the multi-label classification of sexism of any kind(s), and we contribute the largest dataset for sexism categorization. We develop a neural solution for this multi-label classification that can combine sentence representations obtained using models such as BERT with distributional and linguistic word embeddings using a flexible, hierarchical architecture involving recurrent components and optional convolutional ones. Further, we leverage unlabeled accounts of sexism to infuse domain-specific elements into our framework. The best proposed method outperforms several deep learning as well as traditional machine learning baselines by an appreciable margin.

1 Introduction

Sexism, discrimination on the basis of one's sex, prevails in our society in numerous forms, causing immense suffering to women and girls. Online forums have enabled victims of sexism to share their experiences freely and widely by facilitating anonymity and connecting far-away people. A meaningful categorization of these accounts of sexism can play a part in analyzing sexism with a view to developing sensitization programs, systemic safeguards, and other mechanisms against this injustice. Given the rising volume of such information on digital media, automated sexism categorization can aid social scientists and policy makers in combating sexism by conducting such analyses efficiently.

While sexism is detected as a category of hate in some of the hate speech classification work (Badjatiya et al., 2017; Waseem and Hovy, 2016), it does not perform sexism classification. Except the work on categorizing sexual harassment by Karlekar and Bansal (2018), the prior work on classifying sexism assumes the categories to be mutually exclusive (Anzovino et al., 2018; Jha and Mamidi, 2017). Moreover, the existing category sets number between 2 to 5. In this paper, we focus on the new problem of the multi-label categorization of an account of sexism reporting any type(s) of sexism. We create a dataset comprising 13023 accounts of sexism, including first-person accounts from survivors, each tagged with at least one of 23 categories of sexism. The categories were defined keeping in mind the discourse and campaigns on gender-related issues along with potential policy implications, under the guidance of a social scientist. Ten annotators, most of whom have formally studied topics related to gender and/or sexuality, were recruited to label textual accounts of sexism. The accounts are drawn from the Everyday Sexism Project website[1], where voluntary contributors from all over the world document experiences of sexism suffered or witnessed by them. For classification experiments, the categories found in less than 400 accounts in our dataset are appropriately merged with others, resulting in 14 categories.

The rationale for formulating this classification as multi-label is that many experiences inherently involve multiple types of sexism. For instance, *"I*

*The author is also an applied researcher at Microsoft.

[1]https://everydaysexism.com

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1642–1652,
Hong Kong, China, November 3–7, 2019.

overheard a co-worker saying that I should be in more team events and photos because I am pleasing to the eye! Disgusting." is an experience of sexism wherein the victim was subjected to three types of sexism, namely hyper-sexualization, sexual harassment, and hostile work environment.

We develop a novel neural architecture for the multi-label classification of accounts of sexism that enables flexibly combining sentence representations created using models such as BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) with distributional word embeddings like ELMo (Embeddings from Language Models) (Peters et al., 2018) and Global Vectors (GloVe) (Pennington et al., 2014) and a linguistic feature representation through hierarchical convolutional and/or recurrent operations. Leveraging general-purpose models such as BERT for encoding sentences likely makes our model better equipped to capture semantic aspects effectively, since they are trained on substantially larger textual data than the domain-specific labeled data that we have. Moreover, we adapt a BERT model for the domain of instances of sexism using unlabeled data. Embeddings from sentence encoders are complemented by sentence representations built from word embeddings as a function of trainable neural network parameters. We explore multiple ways to deal with the multi-label aspect. The adopted method produces label-wise probabilities directly and simultaneously using shared weights and a joint loss function. Our experimentation finds multiple instances of the proposed framework outperforming several diverse baselines on established multi-label classification metrics.

Our key contributions are summarized below.

- We propose a neural framework for the multi-label classification of accounts of sexism that can combine sentence representations built from word embeddings of different kinds through learnable model parameters with those created using pre-trained models. It yields results superior to many deep learning and traditional machine learning baselines.
- To the best of our knowledge, this is the first work on classifying an account recounting any type(s) of sexism without the assumption of the mutual exclusivity of classes.
- We provide a dataset consisting of 13023 accounts of sexism by survivors and observers annotated with one or more of 23 carefully formulated categories of sexism.

2 Related Work

Substantial work has been directed to hate speech detection in recent years. Since some of it involves the detection of sexism (Badjatiya et al., 2017; Waseem and Hovy, 2016), we review it along with the work on sexism classification.

2.1 Hate Speech Detection

Warner and Hirschberg (2012) identify anti-semitic hate speech using SVM. Gao et al. (2017) perform hate speech detection in a weakly supervised fashion. Nobata et al. (2016) distinguish abusive comments from clean ones through various NLP and embedding-derived features. Burnap and Williams (2016) classify cyber hate with respect to race, disability, and sexual orientation using text parsing to extract typed dependencies. Waseem and Hovy (2016) explore the role of extra-linguistic features along with character n-grams in classifying tweets as racist, sexist or neither. Badjatiya et al. (2017) experiment with various deep learning approaches for the same three-way classification. Zhang and Luo (2018) explore skipped CNN and a combination of CNN and GRU for hate speech detection. Unlike these papers, we seek to categorize accounts of sexism, a specific form of discrimination or hate.

2.2 Sexism Categorization

Anzovino et al. (2018) use features such as (Part of Speech) n-grams and text embeddings for the 5-class categorization of misogynistic language. Jafarpour et al. (2018) classify sexist tweets into one of four categories, which deal with harassment and threats. Jha and Mamidi (2017) experiment with SVM, biLSTM with attention, and fastText to categorize tweets as benevolent, hostile, or non-sexist. Their way of categorizing sexism relates to the manner in which sexism is stated. While it is applicable to all sexist remarks, narrated accounts of sexism may not always capture how the perpetrators stated sexism. Karlekar and Bansal (2018) focus on accounts of sexual harassment, exploring CNN and/or RNN for their 3-class classification. As far as we know, our paper presents the first attempt to categorize accounts involving any type(s) of sexism in a multi-label way. Moreover, we pro-

vide a larger dataset and significantly more extensive or finer-grained categorization scheme than these papers.

3 Dataset Construction

Creating our multi-label sexism account categorization dataset entailed two parts: textual data collection and data annotation. To collect data, we crawled the Everyday Sexism Project website, which receives numerous accounts of sexism from survivors themselves as well as observers. After removing entries with less than 7 words, around 20000 entries were shortlisted for annotation; we prioritized shorter ones and tried to approximate the tag distribution on the website. Though shorter entries were preferred keeping in mind the potential future work of transfer learning to Twitter content, our neural framework is devised in a size-agnostic way.

Under the direction of a social scientist, 23 categories of sexism were formulated taking into account gender-related discourse and campaigns (Dutta and Sircar, 2013; Eccles et al., 1990; Mead, 1963; Menon, 2012) as well as possible impact on public policy. Table 1 provides succinct descriptions for the categories.

We followed a three-phase annotation process to ensure that the categorization of each account of sexism in the final dataset involved the labeling of it by at least two of our 10 annotators, most of whom have studied topics related to gender and/or sexuality formally. The annotators were given detailed guidelines, which evolved during the course of their work. Each annotator was given training, which included a pilot round involving evaluation and feedback. Phase 1 involved identifying one or more textual portions capturing distinct accounts of sexism from an entry obtained from Everyday Sexism Project and subsequently tagging each portion with at least one of the 23 categories of sexism, producing over 23000 labeled accounts. In phases 2 and 3, we sought redundancy of annotations for improved quality, as permitted by the availability of annotators adequately knowledgeable about sexism. Over 21000 accounts were categorized again in phase 2 such that the annotators for phases 1 and 2 were different. The inter-annotator agreement across phases 1 and 2, measured by the average of the Cohen's Kappa (Cohen, 1960; Artstein and Poesio, 2008) scores for the per-category pairs of binary label vectors, is 0.584. Each account for which the label sets annotated across phases 1 and 2 were identical was included in the dataset along with the associated label set. In phase 3, some of the accounts for which there was a mismatch between the phase 1 and phase 2 annotations were selected. For each account, the annotators were presented with only the mismatched categories and asked to select or reject each. Duplicates and records for which the Everyday Sexism Project entry numbers match but the accounts do not fully match were removed at multiple stages. In order to improve the annotation reliability further, some records for which the annotations differed across phases 1 and 2 were discarded based on the annotators involved and sensitivities of the categories, resulting in a multi-label sexism categorization dataset of 13023 accounts. For our automated sexism classification experiments, we merge the categories found in less than 400 records with others as follows, resulting in 14 categories. 'Menstruation-related discrimination' and 'Motherhood-related discrimination' are merged into 'Motherhood and menstruation related discrimination'; 'Mansplaining', 'Gaslighting', 'Religion-based sexism', 'Physical violence (excluding sexual violence)', and 'Other' are merged into 'Other'; 'Pay gap' and 'Hostile work environment (excluding pay gap)' are merged into 'Hostile work environment'; 'Tone policing', 'Moral policing (excluding tone policing)', and 'Victim blaming' are merged into 'Moral policing and victim blaming'; 'Rape' and 'Sexual assault (excluding rape)' are merged into 'Sexual assault'. Our dataset, however, retains all 23 categories. Fig. 1 shows the frequency distribution of the number of labels per account in the dataset, demonstrating the multi-label nature of instances of sexism.

Caution: 1) The category frequencies in our dataset (used for merging categories) do not represent real-world instances of sexism, as they are affected by several factors including the bias of our sampling scheme toward smaller posts and the small size of our dataset relative to the immense degree of prevalence of sexism in the world. 2) Labeling categories of sexism can be complex in many cases. Hence, despite our best efforts, our labeled data may contain inaccuracies or discrepancies. We also recognize that our categorization scheme could be improved.

Table 1: Descriptions of the categories of sexism used in our dataset

Category	Description
Role stereotyping	Socially constructed false generalizations about certain roles being more appropriate for women; also applies to such misconceptions about men
Attribute stereotyping	Mistaken linkage of women with some physical, psychological, or behavioral qualities or likes/dislikes; also applies to such false notions about men
Body shaming	Objectionable comments or behaviour concerning appearance including the promotion of certain body types or standards
Hyper-sexualization (excluding body shaming)	Unwarranted focus on physical aspects or sexual acts
Internalized sexism	The perpetration of sexism by women via comments or other actions
Pay gap	Unequal salaries for men and women for the same work profile
Hostile work environment (excluding pay gap)	Sexism encountered by an employee at the workplace; also applies when a sexist misdeed committed outside the workplace by a co-worker makes working uncomfortable for the victim
Denial or trivialization of sexist misconduct	Denial or downplaying of sexist wrongdoings
Threats	All threats including wishing for violence or joking about it, stalking, threatening gestures, or rape threats
Rape	FBI's expanded definition of rape
Sexual assault (excluding rape)	Any sexual contact without consent; unwanted touching
Sexual harassment (excluding assault)	Any sexually objectionable behaviour
Tone policing	Comments or actions that cause or aggravate restrictions on how women communicate
Moral policing (excluding tone policing)	The promotion of discriminatory codes of conduct for women in the guise of morality; also applies to statements that feed into such codes and narratives
Victim blaming	The act of holding the victim responsible (fully or partially) for sexual harassment, violence, or other sexism perpetrated against her
Slut shaming	Inappropriate comments made about women 1) deviating from conservative expectations relating to sex or 2) dressing in a certain way when it gets linked to sexual availability
Motherhood-related discrimination	Shaming, prejudices, or other discrimination or misconduct related to the notion of motherhood; also applies to the violation of reproductive rights
Menstruation-related discrimination	Shaming, prejudices, or other discrimination or wrongdoings related to periods
Religion-based sexism	Sexist discrimination or prejudices stemming from religious scriptures or constructs
Physical violence (excluding sexual violence)	Domestic abuse, murder, kidnapping, confinement, or other physical acts of violence linked to sexism
Mansplaining	A woman being condescendingly talked down to by a man; also applies when a man gives an unsolicited advice or explanation to a woman related to something she knows well that she disapproves of
Gaslighting	Sexist manipulation of the victim through psychological means into doubting her own sanity
Other	Any type of sexism not covered by the above categories

3.1 Ethical Data Use and Release

We are committed to following ethical practices, which includes protecting the privacy and anonymity of the victims. We only use accounts of sexism and tags from entries on the Everyday Sexism Project website (ESP). The entry titles, which could contain sensitive information related to the names or locations of the victims (or contributors), are not saved or used at all.

Our dataset can be requested for academic purposes only by providing some prerequisites as recommended by an ethics committee and agreeing to certain terms through our website[2]. The requesters who fulfill these conditions will be emailed 1) the data comprising only numerical placeholders and labels, 2) a script that fetches only accounts of sexism from ESP to obtain the account for each placeholder, and 3) the annotation guidelines used. We have devised this method to ensure that if an entry gets removed from ESP by a victim (or contributor), any and all parts of it in our dataset will also be removed.

[2] https://irel.iiit.ac.in/sexism-classification

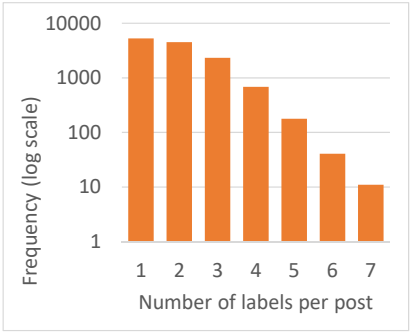

Figure 1: Frequency distribution of #labels per post

4 Sexism Categorization Approach

Given an account of sexism (post), our objective is to predict a list of up to 14 applicable categories of sexism, making this a multi-label multi-class classification task. In this section, we detail our proposed framework, which enables combining sentence representations derived from word embeddings using trainable model parameters with those obtained using general-purpose models. Our architecture is depicted in Fig. 2. We also discuss how we tap unlabeled data and loss functions.

4.1 Sexism Categorization Architecture

Let each post contain a maximum of $|S|$ sentences with a maximum of $|W|$ words per sentence. Ev-

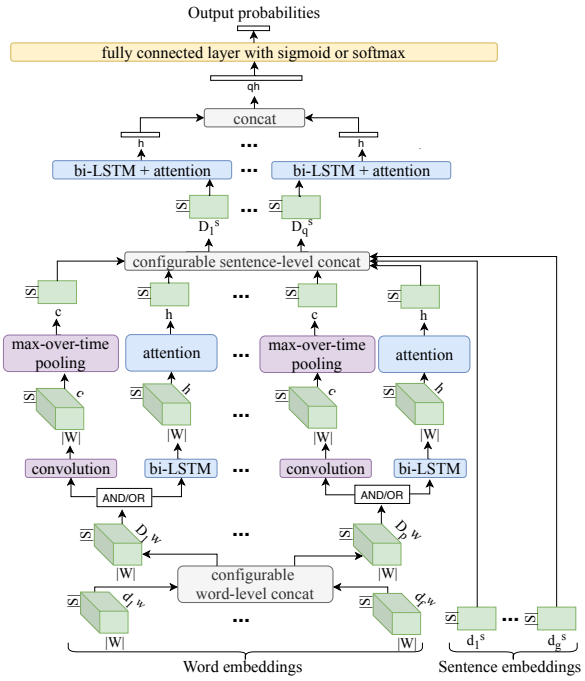

Figure 2: Proposed sexism categorization architecture

ery word (or sentence) can be represented using multiple word (or sentence) embedding methods. Let f (or g) be the number of word (or sentence) embedding methods chosen. Let d_i^w (or d_j^s) be the embedding dimension for the i^{th} (or j^{th}) word (or sentence) embedding scheme. Each post is represented using two kinds of tensors: (a) f tensors $\in \mathcal{R}^{|S|\times|W|\times d_i^w}$ created using different word embeddings, and (b) g tensors $\in \mathcal{R}^{|S|\times d_j^s}$ constructed using different sentence encoders.

First, subsets of the f tensors based on word embeddings are concatenated in a configurable manner (*configurable word-level concat* in Fig. 2), producing p tensors $\in \mathcal{R}^{|S|\times|W|\times D_i^w}$, where D_i^w is the dimension resulting from the i^{th} concatenation. Next, we construct vector representations for the sentences word-embedded in each of the p tensors using CNN-based and/or LSTM-based operations as configured. The CNN-based operations begin with convolutional filters being applied along the word dimension (Kim, 2014) to generate many bigram, trigram and 4-gram based features. This is followed by max-over-time pooling, which picks the largest value for each filter and produces a sentence representing tensor $\in \mathcal{R}^{|S|\times c}$, where c is the total number of convolutional filters used. The LSTM-based components include biLSTM followed by an attention mechanism (Yang et al., 2016) through which the LSTM outputs across time steps are aggregated into a vector representation for each sentence, resulting in a tensor $\in \mathcal{R}^{|S|\times h}$, where h is the bi-LSTM output length. At this stage, we have three types of sentence representing tensors if both CNN-based and RNN-based operations are chosen to be applied on all word embedding tensors: (a) p tensors $\in \mathcal{R}^{|S|\times c}$ from the CNN-based processing, (b) p tensors $\in \mathcal{R}^{|S|\times h}$ from the LSTM-based processing, and (c) g tensors $\in \mathcal{R}^{|S|\times d_j^s}$ obtained using general-purpose sentence encoders.

From these sentence representing tensors, subsets are concatenated to produce q tensors $\in \mathcal{R}^{|S|\times D_j^s}$ (*configurable sentence-level concat* in Fig. 2), where D_j^s is the dimension stemming from the j^{th} concatenation. The sequence of sentence vectors in each of these q tensors is then passed through bi-LSTM followed by attention-based aggregation, producing q representations for a post collectively. These vectors are then concatenated to produce the overall post representation. The final step involves a fully connected layer with a sigmoid or softmax non-linearity depending on the loss function used, generating the output probabilities.

4.2 Word and Sentence Representations

We model a post using both word embeddings and sentence embeddings. We experiment with three distributional word vectors, namely ELMo (Peters et al., 2018), GloVe (Pennington et al., 2014), and fastText (Bojanowski et al., 2017), and a linguistic feature vector. Our linguistic feature representation comprises a variety of features, namely features from the biased language detection work (assertive verb, implicative verb, hedges, factive verb, report verb, entailment, strong subjective, weak subjective, positive word, and negative word) (Recasens et al., 2013), PERMA (Positive Emotion, Engagement, Relationships, Meaning, and Accomplishments) features for both polarities (Schwartz et al., 2016), associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive) from the NRC emotion lexicon (Mohammad and Turney, 2013), and affect (valence, arousal, and dominance) scores (Mohammad, 2018). Missing values are filled with zero for binary features and with the mean for non-binary ones.

We explore the following for creating sentence embeddings: BERT (Devlin et al., 2018), Universal Sentence Encoder (USE) (Cer et al., 2018), and

InferSent (Conneau et al., 2017). Our choice of utilizing these models is warranted by the fact that the corpora that they are trained on are considerably bigger than the textual data that we have for supervised learning and hence likely contain greater semantic diversity.

4.3 Utilizing Unlabeled Data

Models such as BERT are not trained to generate representations tuned to a specific domain. We use over 90000 entries crawled from Everyday Sexism Project's website to tailor a pre-trained BERT model for obtaining more effective representations for our model. After removing the unlabeled entries corresponding to the posts in the test and validation data, we use the rest to tune the BERT parameters using its masked language modeling and next sentence prediction tasks. We henceforth refer to this refined model as tBERT (tuned BERT).

4.4 Loss Function Choice

Since the popular cross-entropy loss is inapt for our multi-label classification task in its standard form, we explore two alternatives. Binary (multi-hot) target vectors are used for both.

4.4.1 Extended Binary Cross Entropy Loss

We adopt an Extended version of the Binary Cross Entropy loss ($EBCE$), formulated as a weighted mean of label-wise binary cross entropy values in order to neutralize class imbalance.

$$L_{EBCE} = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{L}\sum_{j=1}^{L} w_{jy_{ij}}\left\{y_{ij}\log(\hat{p}_{ij}^{\sigma}) + (1-y_{ij})\log(1-\hat{p}_{ij}^{\sigma})\right\} \quad (1)$$

Here, n and L denote the number of samples (posts) and the number of classes respectively. y_{ij} is 1 if label l_j applies to post x_i and 0 otherwise. $\hat{p}_{ij}^{\sigma}$ is the estimated probability of label l_j being applicable to post x_i computed using a sigmoid activation atop the fully connected layer with L output units. The weights for correcting class imbalance w_{jv} are computed as follows.

$$w_{jv} = \frac{n}{2|\{x_i \mid y_{ij} = v, 1 \leq i \leq n\}|} \quad (2)$$

4.4.2 Normalized Cross Entropy Loss

We also experiment with a Normalized variant of the Cross Entropy loss tailored for a multi-label problem configuration also mitigating class imbalance (referred to as NCE).

$$L_{NCE} = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\mathbf{y_i^+}|}\sum_{j=1}^{L} w_j^c\{y_{ij}\log(\hat{p}_{ij})\} \quad (3)$$

Here, $\mathbf{y_i^+}$ is the set of labels applicable to post x_i. $\hat{p}_{ij}$ denotes the estimated probability of label l_j being applicable to post x_i computed through a softmax function. The class imbalance negating weights w_j^c are generated as follows.

$$w_j^c = \frac{n}{\sum_{i=1}^{n}\frac{y_{ij}}{|\mathbf{y_i^+}|}} \quad (4)$$

Unlike in single-label multi-class classification, wherein $\arg\max$ can be applied to the probability vector generated by softmax to make the prediction, we could apply a threshold on probability vector $\hat{\mathbf{p}}_\mathbf{i}$ to find the (potentially) multiple applicable classes for post x_i. Instead of using a fixed, manually tuned threshold-related parameter, we devise an automated method for estimating a per-sample cut-off position. For each sample, we descendingly sort the probability scores, compute the differences between successive (sorted) score pairs, find the index m corresponding to the maximum value in the list of differences, and select the classes corresponding to indices $[1..m]$. Note that when sigmoid (along with $EBCE$ loss) is used instead of softmax, the prediction is made by rounding the probability vector, since it comprises the class-wise binary prediction probabilities.

4.4.3 Discussion on Single-label Transformations

Traditional approaches to multi-label classification include transforming the problem to one or more single-label classification problems. The Label Powerset (LP) method (Boutell et al., 2004) treats each distinct combination of classes existing in the training set as a separate class. The standard cross-entropy loss can then be used along with softmax. This transformative method may impose a greater computational cost than the direct approach using the $EBCE$ loss since the cardinality of the transformed label set may be relatively high. Moreover, LP does not generalize to label combinations not covered in the training set. Another approach based on problem transformation is binary relevance (BR) (Boutell et al., 2004). An independent binary classifier is trained to predict the applicability of each label in this method. This entails training a total of L classifiers, making BR computationally very expensive. Additionally, its performance is affected by the fact that it disregards correlations existing between labels.

5 Experiments

We evaluate the proposed framework against several baselines and provide qualitative and quantitative analyses. Our code is available on GitHub[3]. Our implementation utilizes parts of the code from (Agrawal and Awekar, 2018; Pattisapu et al., 2017; Liao, 2017) and libraries Keras and Scikit-learn (Pedregosa et al., 2011). We reserve 15% of the data for testing and validation each.

5.1 Evaluation Metrics

Owing to the multi-label nature of this classification, standard metrics used in single-label multi-class classification are unsuitable. We adopt established example (instance) based metrics, namely F1 (F_I) and accuracy (Acc_I), and label-based metrics, namely F1 macro (F_{macro}) and F1 micro (F_{micro}) used in multi-label classification (Zhang and Zhou, 2014).

5.2 Baselines

Random

Labels are selected randomly as per their normalized frequencies in the training data for each test sample.

Traditional Machine Learning (ML)

We experiment with Support Vector Machine (SVM), Random Forests (RF), and Logistic Regression (LR). The features explored include TF-IDF on character n-grams (1-5 characters), TF-IDF on word unigrams and bigrams, the mean of the ELMo vectors for the words in a post, and the composite set of features similar to (Anzovino et al., 2018) comprising n-gram based, POS-based, and doc2vec (Le and Mikolov, 2014) features, the post length, and the adjective count.

LSTM-based Architectures

biLSTM: The word embeddings for all words in a post are fed to bidirectional LSTM.

biLSTM-Attention: Same as biLSTM but with the attention mechanism by Yang et al. (2016).

Hierarchical-biLSTM-Attention: For the words in each sentence, the word embeddings are passed through biLSTM with attention to create a sentence embedding. These sentence embeddings are in turn fed to another instance of biLSTM with attention. This broadly follows the architecture proposed for document classification by Yang et al. (2016) with GRUs replaced with LSTMs.

[3] https://github.com/pulkitparikh/sexism_classification

Sentence embeddings with biLSTM-Attention: Sentence representations generated using a generic encoder (BERT using bert-as-service (Xiao, 2018), USE, and InferSent) are passed through biLSTM with attention.

CNN and CNN-LSTM based Architectures

CNN-Kim: Similar to (Kim, 2014), this involves applying convolutional filters followed by max-over-time pooling to the word vectors for a post.

C-biLSTM: In this variant of the C-LSTM architecture (Zhou et al., 2015) somewhat related to an approach used by Karlekar and Bansal (2018) for multi-label sexual harassment classification, after applying convolution on the word vectors for a post, the feature maps are stacked along the filter dimension to create a sequence of window vectors, which are then fed to biLSTM.

CNN-biLSTM-Attention: For each sentence, convolutional and max-over-time pooling layers are applied on the embeddings of its words. The resultant sentence representations are put through bi-LSTM with the attention mechanism. This approach is similar to (Wang et al., 2016) with the attention scheme from (Yang et al., 2016) added.

The architectures of the deep learning baselines have a fully connected layer with the sigmoid or softmax non-linearity (depending on the loss function used) at the end.

5.3 Results

Table 2 shows results produced using traditional ML methods (SVM, RF, and LR) across four different feature sets (word n-grams, character n-grams, averaged ELMo vectors, and composite features). We use Label Powerset for these methods, since the direct (non-transformative) formulation cannot be used with them. Among these combinations, logistic regression with averaged ELMo embeddings as features performs the best.

Table 3 contains results for the random and deep learning baselines and different variants of the proposed framework. For each method, the average over three runs is reported for each metric. We find ELMo to be better than GloVe and fastText for word embeddings across multiple baselines and hence show only ELMO-based results for the baselines. We report all results with the EBCE loss; the NCE loss produced inferior results across multiple methods. For our framework, s() denotes sentence-level concatenation; wl() de-

Table 2: Results with traditional machine learning (Label Powerset)

Features⟶	Word n-grams				Character n-grams				Averaged ELMo vectors				Composite features			
Classifier↓	F_I	F_{macro}	Acc_I	F_{micro}	F_I	F_{macro}	Acc_I	F_{micro}	F_I	F_{macro}	Acc_I	F_{micro}	F_I	F_{macro}	Acc_I	F_{micro}
SVM	0.448	0.373	0.324	0.410	0.449	0.374	0.331	0.416	0.546	0.430	0.431	0.500	0.178	0.094	0.116	0.174
LR	0.357	0.315	0.236	0.349	0.357	0.311	0.230	0.352	**0.595**	**0.479**	**0.478**	**0.549**	0.438	0.370	0.311	0.421
RF	0.531	0.398	0.438	0.476	0.395	0.205	0.325	0.349	0.375	0.164	0.305	0.331	0.460	0.311	0.380	0.415

notes word level concatenation and LSTM-based processing; wc() denotes word level concatenation and CNN-based processing. We note that the results are reported for only some of the many instances that can arise from our configurable architecture. Our framework provides the ability to explore different configurations such as those with multiple s() operations, depending on the problem at hand.

We observe the following: (1) The random baseline performs poorly, confirming the complexity of the problem. (2) biLSTM-Attention and Hierarchical-biLSTM-Attention are the two best baselines. (3) Several variants of the proposed framework outperform all baselines. Based on F_I and F_{macro}, our best method is s(wl(ELMo), wl(GloVe), tBERT), though adding linguistic features (Ling) to it slightly improves some metrics. (4) BERT tuned on unlabeled instances of sexism (tBERT) works better than the vanilla BERT counterpart and other sentence encoders. (5) Combining tBERT sentence representations with those generated from ELMo word vectors using biLSTM with attention works better than using either individually. (6) Along with tBERT, concatenating ELMo and GloVe at the sentence level (s(wl(ELMo), wl(GloVe), tBERT)) is better than concatenating them at the word level (s(wl(ELMo, GloVe), tBERT)) while processing word vectors using biLSTM with attention. (7) The LSTM based processing of word embeddings produces better results than the CNN based counterpart.

Table 3: Results for the proposed methods and deep learning baselines (using ELMo embeddings) with the EBCE loss and Random

	Approach	F_I	F_{macro}	Acc_I	F_{micro}
Baselines	Random	0.042	0.141	0.027	0.193
	biLSTM	0.697	0.616	0.563	0.658
	biLSTM-Attention	0.728	0.650	0.601	0.688
	Hierarchical-biLSTM-Attention	0.725	0.650	0.604	0.688
	BERT-biLSTM-Attention	0.656	0.555	0.502	0.611
	USE-biLSTM-Attention	0.628	0.549	0.468	0.594
	InferSent-biLSTM-Attention	0.418	0.37	0.274	0.399
	CNN-biLSTM-Attention	0.714	0.628	0.586	0.671
	CNN-Kim	0.701	0.622	0.574	0.669
	C-biLSTM	0.708	0.631	0.583	0.674
Proposed methods	tBERT-biLSTM-Attention	0.688	0.589	0.539	0.644
	s(wl(ELMo), tBERT)	0.747	0.675	0.628	0.710
	s(wl(ELMo, GloVe), tBERT)	0.743	0.667	0.618	0.703
	s(wc(ELMo), wc(GloVe), tBERT)	0.738	0.654	0.614	0.698
	s(wl(ELMo), wl(GloVe), tBERT)	**0.756**	**0.684**	**0.635**	**0.715**
	s(wl(ELMo), wl(GloVe), tBERT, USE)	0.753	0.673	0.632	0.715
	s(wl(ELMo), wl(GloVe), wl(Ling), tBERT)	**0.753**	**0.685**	**0.636**	**0.718**
	s(wc(ELMo), wl(ELMo), wc(GloVe), wl(GloVe), tBERT)	0.741	0.664	0.625	0.705

The pre-processing steps that we perform for all deep learning methods include removing certain non-alpha-numeric characters and extra spaces, lower-casing, and zero-padding input tensors as appropriate. While breaking a post into sentences, each sentence containing more than 35 words is split into multiple sentences, ensuring the maximum sentence length of 35 words.

Using experiments on a validation set, which was merged into the training set during the test runs, for each method, we choose the values of three hyper-parameters: the LSTM dimension, the attention dimension, and the number CNN filters for kernel sizes 2, 3, and 4 each. The values used for instances of our framework and the deep learning baselines are provided in Table 5.

We employ 0.25 dropouts after each input and before the final, fully connected layer. The learning rate was set to 0.001 and the number of epochs to 10. We use a batch size of 64. These fixed parameters were kept unchanged across methods.

The hyper-parameter values for the traditional ML methods are as follows. For SVM, soft margin (C) is set to 1.0. For RF (Random Forest), the number of estimators is 100. For extracting character and word n-grams, the maximum number of features used, word n-gram range, and character n-gram range are 10000, (1,2), and (1,5) respectively. For SVM and LR (Logistic Regression), we apply class imbalance correction.

For tapping unlabeled data, we pre-train the 'BERT-Base, Uncased' model[4] for 100000 steps with a batch size of 25. For vannila BERT, we use the bigger 'BERT-Large, Uncased' model, which we could not use for pre-training because of computational constraints. For generating GloVe word embeddings, we use the 840B-token, cased model.

Table 4 lists accounts of sexism from the test set for which our best method made the right predictions but the best baseline did not, along with the labels. It also highlights the top two words per sentence based on the word-level attention weights for wl(ELMo) and wl(GloVe) combined through element-wise max operations. For the first ac-

[4] https://github.com/google-research/bert

Table 4: Attention analysis for some test samples

Account of sexism	Two most-attended-to words / sentence	Labels
am in social services. my male coworker and I have the same job title. Everyone assumes I am his assistant or an intern. It's not just clients who assume he's my boss, it's other agencies. The same ppl who think nothing calling me honey at work.	(services, am), (coworker, title), (intern, assistant), (boss, agencies), (honey, me)	Role stereotyping, Hostile work environment, Sexual harassment (excluding assault)
being told I should take cat calls as compliments by my father	(compliments, cat)	Denial or trivialisation of sexist misconduct, Sexual harassment (excluding assault)
I didn't appreciate it when my own father walked into the house one day while I was doing laundry and told me that "it's nice to see you finally doing women's work."	(womens, work)	Role stereotyping, Moral policing and victim blaming
Referred to as 'not a girl' because I have short hair and don't wear noticeable makeup.	(makeup, hair)	Attribute stereotyping, Body shaming
At our school girls are forbidden to wear tight trowsers or remove their blazers- in fear of distracting the boys.	(tight, trowsers)	Moral policing and victim blaming, Hyper-sexualization (excluding body shaming)

Table 5: Tuned hyper-parameter values for the deep learning baselines (ELMo embeddings) and proposed methods with the EBCE loss

Approach	LSTM dimension	Attention dimension	CNN Filters of each kernel size
Hierarchical-biLSTM-Attention	300	400	N.A.
CNN-biLSTM-Attention	300	400	100
BERT-biLSTM-Attention	200	500	N.A.
USE-biLSTM-Attention	300	600	N.A.
InferSent-biLSTM-Attention	100	200	N.A.
C-biLSTM	300	N.A.	150
biLSTM-Attention	200	300	N.A.
biLSTM	300	N.A.	N.A.
CNN-Kim	N.A.	N.A.	150
tBERT-biLSTM-Attention	300	600	N.A.
s(wl(ELMo), tBERT)	300	600	N.A.
s(wl(ELMo, GloVe), tBERT)	100	100	N.A.
s(wl(ELMo), wl(GloVe), tBERT)	100	200	N.A.
s(wl(ELMo), wl(GloVe), tBERT, USE)	300	600	N.A.
s(wl(ELMo), wl(GloVe), wl(Ling), tBERT)	300	600	N.A.
s(wc(ELMo), wc(GloVe), tBERT)	300	600	100
s(wc(ELMo), wl(ELMo), wc(GloVe), wl(GloVe), tBERT)	300	500	100

Table 6: Performance variation across #labels per post

#Labels per post	F_I	F_{macro}	Acc_I	F_{micro}
1	0.729	0.527	0.632	0.637
2	0.754	0.675	0.631	0.722
3	0.781	0.721	0.652	0.764
4	0.743	0.722	0.604	0.735
5	0.739	0.592	0.592	0.735

count of sexism, our model produces words like "intern", "assistant" and "boss", associated with role stereotyping, among the top two words across sentences. Likewise, "honey" related to sexual harassment and "boss", "coworker", and "services" related to hostile work environments also surface. Moreover, the top two sentences based on the sentence-level attention weights of our model are the last two, which evidence all category labels. For other posts too, the model produces category-relevant top two words per sentence; "womens" and "work" relate to role stereotyping and moral policing; "tight" and "trowsers" relate to hyper-sexualization and moral policing; "makeup" relates to attribute stereotyping; "hair" from "short hair" relates to body shaming; "cat" from "cat calls" relates to sexual harassment; "compliments" from "take cat calls as compliments" relates to denial or trivialization of sexist misconduct.

Table 6 shows results for one run of our best method across different numbers of labels per post (1 to 5). Entries for values 6 and 7, which have less than 10 associated test samples, are omitted. The best results are observed for values 2 to 4, suggesting that our approach performs better on multi-label samples.

6 Conclusion

We explored classifying an account reporting any kind(s) of sexism such that the categories can co-occur. We developed a neural framework that outperforms many deep learning and traditional ML baselines for this multi-label sexism classification. Moreover, we provided the largest dataset for sexism classification, linked with 23 categories. Directions for future work include devising approaches that perform sexism classification more accurately, enhancing the categorization scheme, and developing other ways to help counter sexism. We hope that this paper will give rise to further work aimed at fighting sexism.

7 Acknowledgments

We are really grateful to Everyday Sexism Project. We also thank Bhavapriya Thottakad, Himakshi Baishya, Aanchal Gupta, Subhashini R, Shalini Pathi, Hadia Ahsan, and Reshma Kalpana, without whose efforts in analyzing accounts of sexism this paper would not have materialized. We are grateful to Ponnurangam Kumaraguru, too, for his time and inputs.

References

Sweta Agrawal and Amit Awekar. 2018. Deep learning for detecting cyberbullying across multiple social media platforms. In *European Conference on Information Retrieval*, pages 141–153. Springer.

Maria Anzovino, Elisabetta Fersini, and Paolo Rosso. 2018. Automatic identification and classification of misogynistic language on twitter. In *International Conference on Applications of Natural Language to Information Systems*, pages 57–64. Springer.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760. International World Wide Web Conferences Steering Committee.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. 2004. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771.

Pete Burnap and Matthew L Williams. 2016. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data Science*, 5(1):11.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Debolina Dutta and Oishik Sircar. 2013. India's winter of discontent: Some feminist dilemmas in the wake of a rape. *Feminist Studies*, 39(1):293–306.

Jacquelynne S Eccles, Janis E Jacobs, and Rena D Harold. 1990. Gender role stereotypes, expectancy effects, and parents' socialization of gender differences. *Journal of social issues*, 46(2):183–201.

Lei Gao, Alexis Kuppersmith, and Ruihong Huang. 2017. Recognizing explicit and implicit hate speech using a weakly supervised two-path bootstrapping approach. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 774–782.

Borna Jafarpour, Stan Matwin, et al. 2018. Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 107–114.

Akshita Jha and Radhika Mamidi. 2017. 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*, pages 7–16.

Sweta Karlekar and Mohit Bansal. 2018. Safecity: Understanding diverse forms of sexual harassment personal stories. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2805–2811.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Richard Liao. 2017. textclassifier. https://github.com/richliao/textClassifier.

Margaret Mead. 1963. *Sex and temperament in three primitive societies*, volume 370. Morrow New York.

Nivedita Menon. 2012. *Seeing like a feminist*. Penguin UK.

Saif Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee.

Nikhil Pattisapu, Manish Gupta, Ponnurangam Kumaraguru, and Vasudeva Varma. 2017. Medical persona classification in social media. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 377–384. ACM.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1650–1659.

H Andrew Schwartz, Maarten Sap, Margaret L Kern, Johannes C Eichstaedt, Adam Kapelner, Megha Agrawal, Eduardo Blanco, Lukasz Dziurzynski, Gregory Park, David Stillwell, et al. 2016. Predicting individual well-being through the language of social media. In *Biocomputing 2016: Proceedings of the Pacific Symposium*, pages 516–527. World Scientific.

Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. 2016. Dimensional sentiment analysis using a regional cnn-lstm model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 225–230.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Han Xiao. 2018. bert-as-service. https://github.com/hanxiao/bert-as-service.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837.

Ziqi Zhang and Lei Luo. 2018. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, pages 1–21.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*.