CSCE - 638

Identification & Classification of Toxic Comments

Abhay Kumar Singh, Mukund Srinath Heragu,
Rizu Jain, Vindhya Ningegowda

What are Toxic Comments?

TEXAS A&M UNIVERSITY

Toxic comments are statements that may intentionally or unintentionally hurts a person’s sentiments.

Nonsense? kiss off, geek. What I said is true. I’ll have your account terminated. — ❌ TOXIC

"Ban one side of an argument by a bullshit nazi admin and you get no discussion because the islamist editors feel they ""won""." — ❌ TOXIC ❌ OBSCENE ❌ INSULT

Why can you put English for example on some players but others people don't like it - why? — ✅ SAFE

Dataset

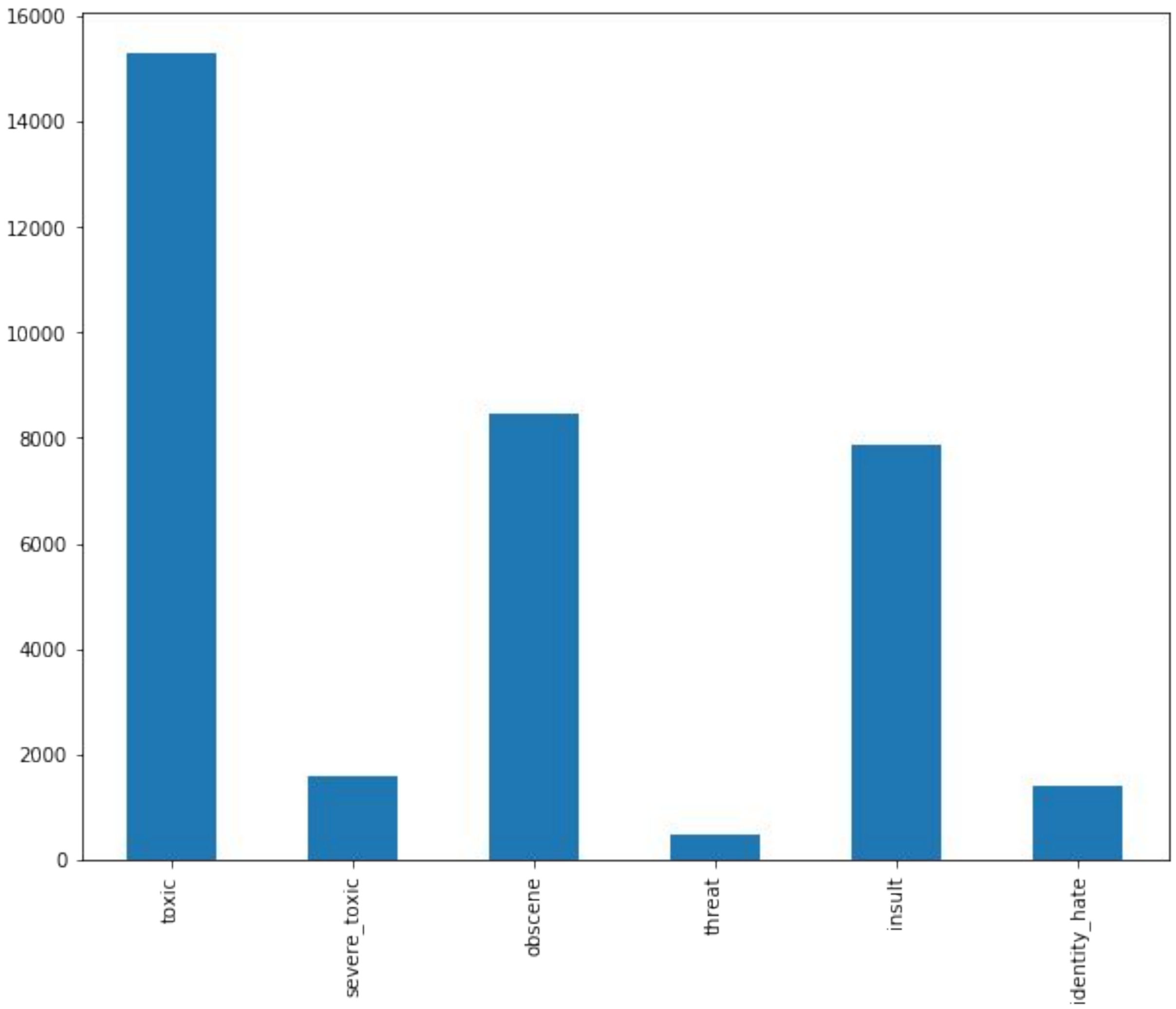
The dataset was taken from a Kaggle Competition conducted by Jigsaw/Conversation AI.

Dataset consists of a comment text in each row with 6 labels namely - toxic, severe_toxic, obscene, threat, insult, identity_hate

id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
000113f07ec002fd	hey man, i'm really not trying to edit war. it...	0	0	0	0	0	0
0002bcb3da6cb337	cocksucker before you piss around on my work	1	1	1	0	1	0
00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0
0005c987bdfc9d4b	hey... what is it..\n@ | talk .\nwhat is it......	1	0	0	0	0	0
0007e25b2121310b	bye! \n\ndon't look, come or think of comming ...	1	0	0	0	0	0

Distribution of labels in the dataset

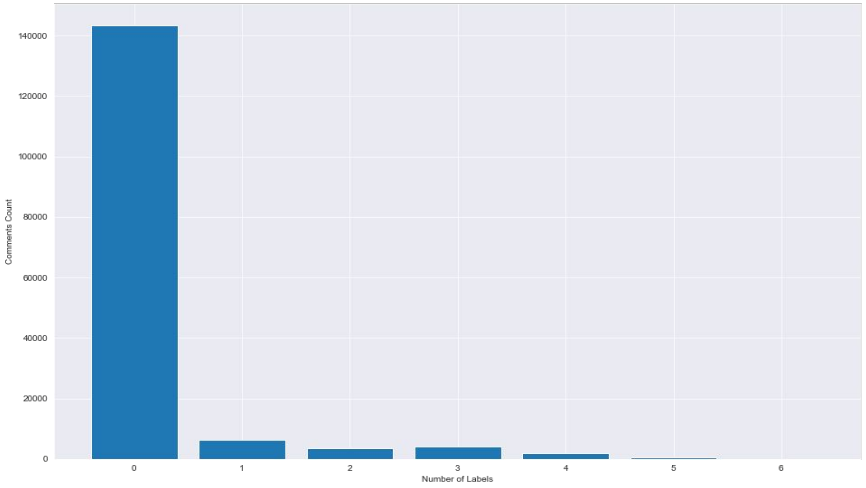
The image on the right shows the frequency of each label in the dataset, with "toxic" having the highest frequency and threat having the lowest frequency.

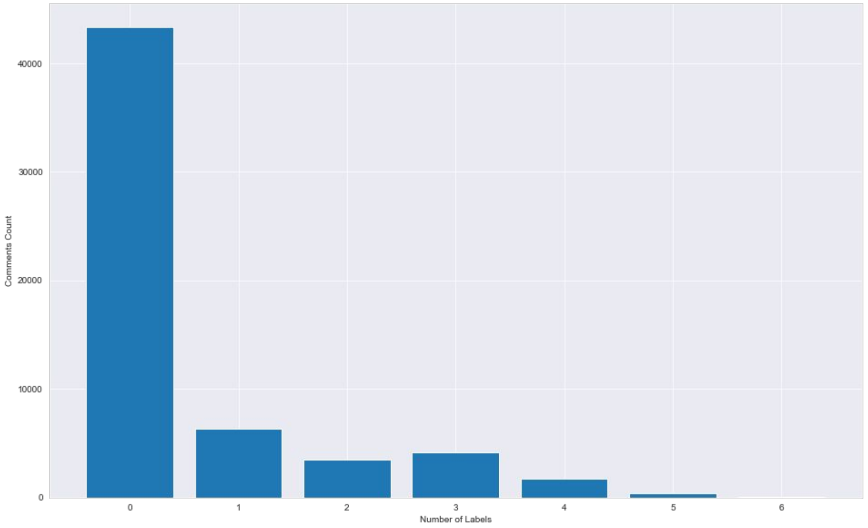


Data Skewness

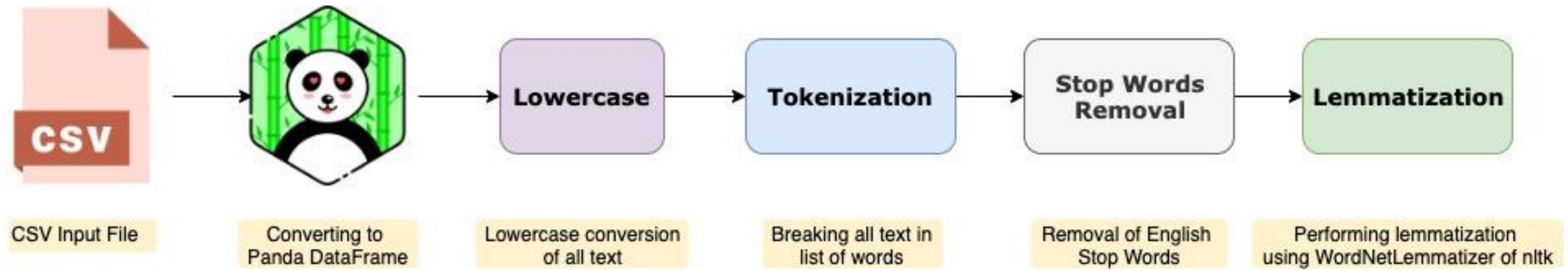
- Since a lot of the data that we have is skewed (having no labels for the comments), running the different models on the data led to a high accuracy value but a low f1 score value.

- In order to combat this problem, we undersampled the dataset by removing 67% of all data that do not contain any labels using random sampling





Preprocessing of Data



Binary Classification

TEXAS A&M UNIVERSITY

For Binary Classification, we created models for Toxic label and used following Word Embedding Techniques -

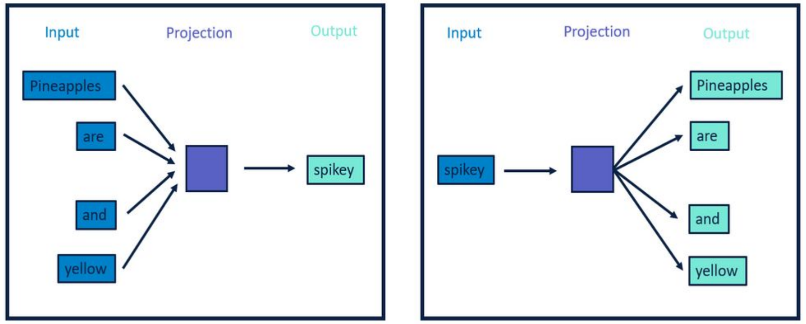
- TF-IDF (Term Frequency–Inverse Document Frequency)

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF
Term x within document y

$tf_{x,y}$ = frequency of x in y
df_x = number of documents containing x
N = total number of documents

- Word2Vec



CBOW　　Skip-gram

Binary Classification - ML Algorithms

Once the data preprocessing and word embedding is done, we are ready to train our model. Models were trained using following Machine Learning Algorithms -

- Naive Bayes
- Logistics Regression (Ridge,Lasso)
- Support Vector Machine (SVM)

Binary Classification Results

ML Algorithm	Word Embedding Technique	Testing Accuracy	Testing F1 Score	Testing ROC-AUC
Naive Bayes	TF-IDF	0.885	0.776	0.851
	Word2Vec	Word2Vec vectors can contain -ve values but Naive Bayes don't accept -ve values		
Logistics Regression	TF-IDF (Lasso Regularization)	0.912	0.823	0.874
	Word2Vec (Ridge Regularization)	0.859	0.687	0.776
SVM	TF-IDF (Linear Kernel with C=1)	0.906	0.799	0.849
	Word2Vec (RBF Kernel with C=10)	0.876	0.722	0.796

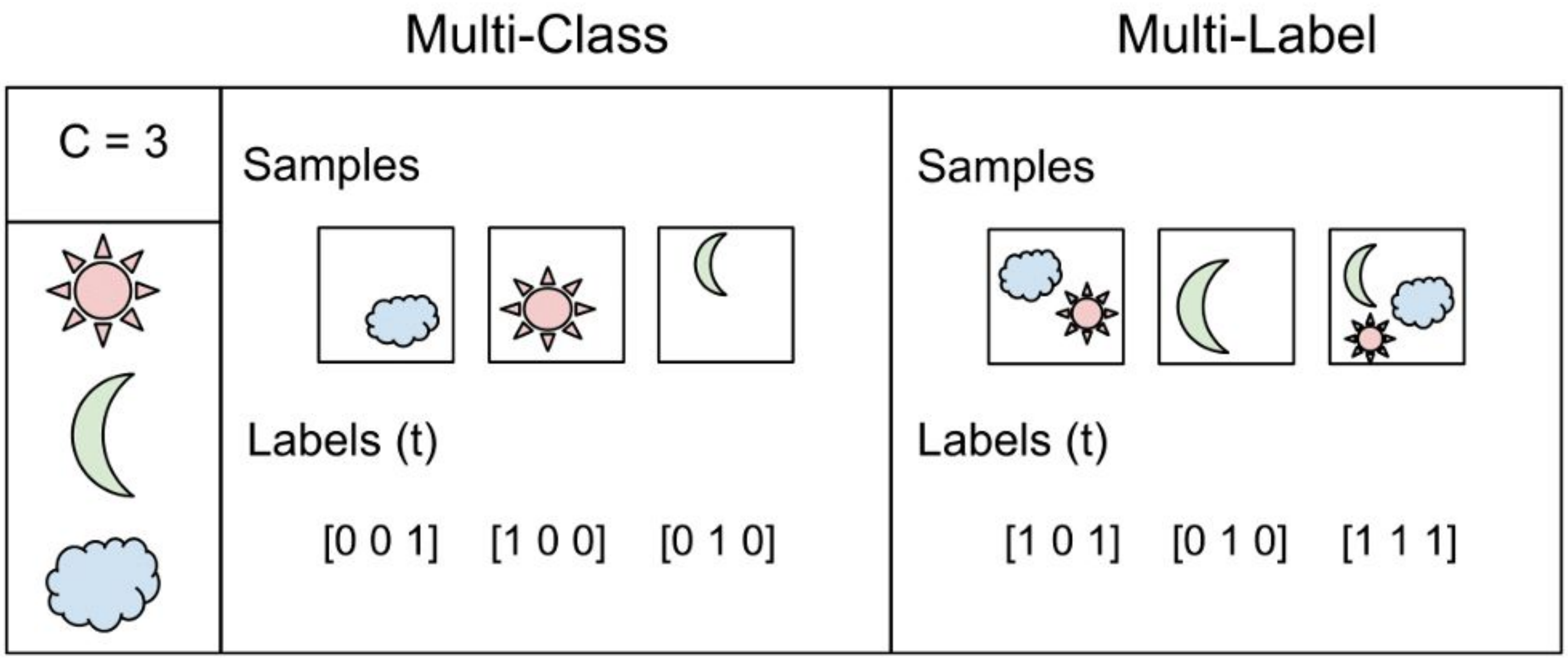
Binary Classification Results

TEXAS A&M UNIVERSITY

- For Logistics Regression, we performed Cross Validation over multiple values of C and regularization techniques. For TF-IDF, Lasso Regression gave us better metrics and for Word2Vec, Ridge Regularization gave us better metrics. Above scores are for Lasso Regularization.
- In SVM, we performed Cross Validation over Polynomial and RBF kernel with other hyper-parameters. For TF-IDF, SVM with Polynomial Kernel of degree 1 (Linear Kernel) with C = 1 performed best and for Word2Vec, SVM with RBF kernel . Above scores are for this SVM model.

Multilabel Classification

For Multilabel Classification, we created models using GloVe and FastText word embedding techniques and trained our models using LSTM.



LSTM

- Long Short-Term Memory is an artificial recurrent neural network architecture used in the field of deep learning. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).

- A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

- We implement LSTM to solve our multi-label classification problem by using Keras.

GloVe

- GloVe (short for Global Vectors) is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space

- In our implementation we have used pre-trained word vectors file (of 6B tokens and 100 dimension vectors) trained on Wikipedia and Gigaword 5.

FastText

TEXAS A&M UNIVERSITY

- FastText is a word embedding method created by Facebook's AI Research (FAIR) lab. Each word is represented as n-gram of characters. It captures meaning of shorter words and allow embedding to understand prefixes/suffixes. If a word was not seen during training, it can be broken down into n-grams to get its word embedding, inherently making it perform better in case of rare words.

- Our CNN model used pre-trained 2 Million word vectors trained on Common Crawl (600B tokens).

Multilabel Classification Results

Algorithm	Word Embedding Technique	Testing Accuracy	Testing ROC-AUC
LSTM	Glove	0.950	0.927
	FastText	0.956	0.949

- Our CNN model implemented bidirectional GRU-LSTM-pooling using pre-trained FastText embeddings. Bidirectional LSTM and GRU helped capture context in both directions of the comments and FastText embeddings performed better because of the n-gram subwords information aiding in better classification of toxic words which are rare in general text corpus.

Conclusion

TEXAS A&M UNIVERSITY

- Out of all the models we implemented, for multi-label classification, LSTM with FastText ran the best and for binary classification, Logistic Regression with Lasso Regularization performed the best.

Improvements:

- Improvements to solutions to the given problem have been obtained using BERT. BERT is a method of pre-training language representations, meaning that we train a general-purpose "language understanding" model on a large text corpus (like Wikipedia), and then use that model for downstream NLP tasks that we care about (like question answering). BERT can be used to perform a wide range of NLP tasks ranging from classifying languages to classifying toxic comments.

Thank You!

Questions?