Shared Task Information Science - Baseline Design

Murali Manohar
S5397294

Karlo Slot
S4031970

Sanne Weering
S4127579

Amber Chen
S3333302

1 Introduction

SemEval 2023, more precisely the task of explainable Detection of Online Sexism (EDOS) (Guest et al., 2021) is a shared task focused on online sexism. We will be participating in this shared task and in this article we present our baselines for the SemEval 2023 task and discuss various preprocessing and feature selection techniques while implementing them.

2 Preprocessing

To ensure that the SemEval 2023 training data is used to its full potential, we have used several methods to clean the data for our baselines.

First, we used Python's `.lower()` method to change all characters in the data entries to lowercase. This was done to reduce possible noise in the data and to ensure that tokens with capitalization, i.e. words at the start of a sentence, are treated the same as its non-capitalized version. One drawback is that named entities are treated the same as normal tokens. For example, "Trump" as a last name and as a verb would be mapped to the same token.

Second, we interpret this task as a textual problem focused on hate speech. Therefore, the choice has been made to remove non-words, such as URLs, interpunction and non-alphanumeric characters. This was accomplished using the regular expressions `"http\S+"` (URLs) and `"[^\w\s]"` (non-word/non-alphanumeric characters, interpunction).

Third, we tokenized and lemmatized the texts using NLTK[1]. For the lemmatization, we used NLTK's `WordNetLemmatizer()`[2], while we used NLTK's `word_tokenize()`[3] function for tokenization.

3 Baselines

To estimate the difficulty of SemEval 2023 task, we start by implementing the baselines, most frequent classifier and SVM classifier. In addition to the descriptions of the baselines, we briefly talk about the data splitting and the features used.

1. **Most frequent Classifier:** We start with a simple baseline, where we always predict an instance as the most frequent class label in the training data. While this approach may seem redundant and non-robust, this is useful in the imbalanced scenarios. For example, if a class occurs 90% of the time in the training data, predicting all the test instances as the same class would result in 90% accuracy. This proves to be a strong baseline to beat for the upcoming complex machine learning approaches.

Task	Baseline	Prec.	Rec.	F1	Acc.
A	Maj.	37.98	50.00	43.17	75.95
	SVM + TF	94.26	91.84	**92.97**	95.00
	SVM + TFIDF	94.55	91.40	92.84	94.95
B	Maj.	15.11	20.00	17.22	75.57
	SVM + TF	92.70	78.53	84.65	93.81
	SVM + TFIDF	94.67	77.89	**84.93**	93.48

Table 1: Table depicting the macro-average precision (Prec.), macro-average recall (Rec.), macro-average F1-score (F1) and accuracy (Acc.) of all baseline models per task. Note that, TF refers to the term frequencies and TFIDF refers to the term-frequency-inverse-document-frequency score

2. **Support Vector Machines:** As a second baseline, we evaluate Support Vector Machine (SVM). (Cortes and Vapnik, 1995) The basic notion of this algorithm is to divide the data points into different classes by identifying a hyperplane which maximizes the distance between the nearest data points. We conjecture that the n-grams can help the model to distinguish between named entities, negation, etc. Therefore, we convert comments into n-grams (unigrams and bigrams) and convert them into features by either counting the term frequencies or TF-IDFs. (Wikipedia, 2022)

The dataset is split into 70:15:15 ratio for train, dev and test splits. We used `Stratified Shuffle Split`[4] to split the data in a way that preserves the percentage of samples for each class in each split. As we were not performing hyperparameter search for SVM, we merged train and dev splits into a bigger training split with 85 percent instances of the whole dataset.

4 Results

In table 1, it can be seen that the majority-based baselines performs poorly, since the macro-average F1-score for Task A is 43.17, where the majority label is 'not sexist', and for Task B is 17.22, where the majority label is 'none'.

In the aforementioned table, it can also be seen that the SVM trained on term frequencies (TF) performs better with a macro-average F1-score of 92.97 on Task A. And on Task B, SVM trained on TF-IDF performs slightly better than TF with a F1-score of 84.93 on Task B respectively.

Since the scores of the baselines are reasonable high, it would be tough to beat these baselines scores.

[1] NLTK Documentation

[2] NLTK's `WordNetLemmatizer()` Documentation

[3] NLTK's `word_tokenizer()` Documentation

[4] sklearn's `Stratified Shuffle Split`

References

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks.

Ella Guest, Bertie Vidgen, Alexandros Mittos, Nishanth Sastry, Gareth Tyson, and Helen Margetts. 2021. An expert annotated dataset for the detection of online misogyny.

Wikipedia. 2022. tf–idf.