Report of Toxic Spans Detection

Name: Archana Kalburgi

Introduction

Background of the task

Toxic span detection is a task related to identifying toxic language in online text. Toxic language refers to language that is harmful, abusive, or offensive in nature, and can include hate speech, harassment, and other forms of abusive behavior. The goal of toxic span detection is to identify and classify such language in order to improve the safety and inclusivity of online platforms.

Description

SemEval's task on toxic span detection involves building a machine learning model to identify toxic spans of text within a given dataset of online comments. A toxic span is defined as a contiguous span of text that is toxic, as determined by annotators. The task involves training a model to classify each span of text as toxic or not toxic, and to assign a label indicating the type of toxic language present (e.g. hate speech, harassment, etc.).

Potential applications

Some potential applications: Toxic span detection has a number of potential applications, including improving the safety and inclusivity of online platforms by identifying and moderating toxic content, and helping to mitigate the negative effects of toxic language on individuals and communities. It can also be used to inform research on toxic language and its impacts, and to develop strategies for combating it.

Why it is challenging

Toxic span detection is a challenging task for a number of reasons. One of the main challenges is the large amount of variation in toxic language, as it can take many forms and be expressed in a variety of ways. Additionally, toxic language can be difficult to detect due to its often subtle and nuanced nature, and it may be embedded within otherwise benign text. Finally, there are often cultural and linguistic differences that can make it difficult to identify toxic language in text from different regions or languages.

Problem Formulation

Input

The input to the toxic span detection task is a dataset of online comments, each of which is a sequence of words (i.e. a string of text). Each comment is annotated with one or more spans of toxic text, if present.

Output

The output of the toxic span detection task is a classification of each span of text within the input comments as either toxic or not toxic, and a label indicating the type of toxic language present (e.g. hate speech, harassment, etc.).

Task type

Toxic span detection is a task in natural language processing (NLP) that involves identifying spans of text (i.e., sequences of words) that are toxic or offensive. This can be approached as a sequence labeling problem, where each word in the text is assigned a label indicating whether it is part of a toxic span or not.

Given a dataset of N online comments, each represented as a sequence of words X_i (where 1 <= i <= N), the goal is to predict a label Y_i for each span of text within the comment, where Y_i can take on one of K possible values, corresponding to the classes of toxic or not toxic and the different types of toxic language.

The task can be formalized as a supervised learning problem, where the model is trained on a labeled dataset of input-output pairs(X_i, Y_i) and then tested on unseen comments to evaluate its performance.

Method

Data preprocessing

In the data preprocessing phase, the input dataset of online comments is cleaned and prepared for use in training and evaluating the model. The nltk library provides a range of tools for working with natural language data, and these could be used to perform tasks such as removing punctuation, lowercasing all words, and tokenizing the text into individual words and word sequences.

The resulting tokens are then indexed using a vocabulary built from the GloVe Twitter word embeddings, which would provide a fixed-length numerical representation for each word in the dataset. Any missing or corrupted data is handled using masking technique.

Any necessary data augmentation or sampling techniques can be applied to balance the dataset or increase its size.

Model design

For this task I have used the BertClassifier a type of machine learning model that is based on the BertModel, which is a version of the BERT (Bidirectional Encoder Representations from Transformers) model and has been trained on a large dataset of text and fine-tuned for various natural language processing (NLP) tasks.

To use a BertClassifier for toxic span detection I first initialized an instance of the BertModel by calling BertModel.from_pretrained('bert-base-uncased'). This will create a BertModel instance that is initialized with the weights of a pre-trained BERT model, BertTokenizer.from_pretrained('bert-base-uncased', do_lower_case=True) that has been trained on lowercased text.

After initializing the BertModel, it is used as the basis for a BertClassifier by passing it as an argument to the BertClassifier constructor, like this: BertClassifier(BertModel.from_pretrained('bert-base-uncased')).

To train the BertClassifier model for toxic span detection, I have provided it with a labeled dataset of text examples where the toxic spans have been annotated, and use Adam optimizer to optimize the model's parameters.

Number of trainable parameters : **109531521**

Loss function

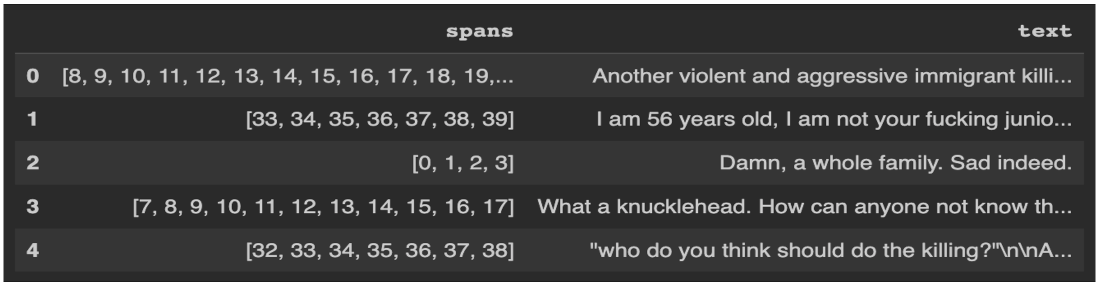
I have used the Binary Cross-Entropy Loss (BCELoss) as the loss function, which is a common choice for binary classification tasks like toxic span detection.

Experiments

Data statistics

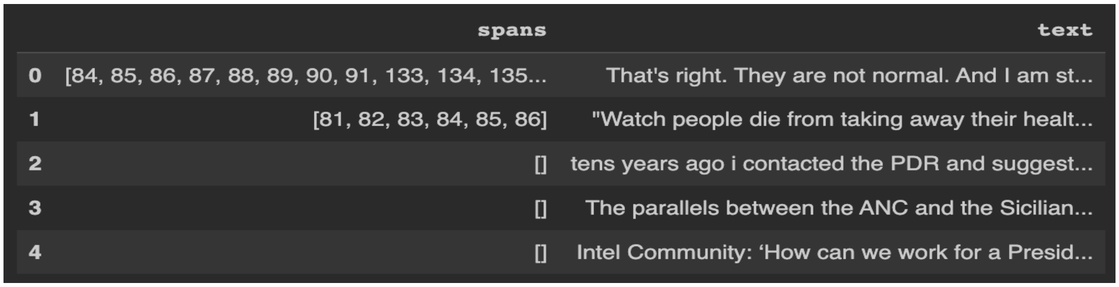
The training set contains 7939 rows and 2 columns, while the test set contains 2000 rows and 2 columns.

Train set sample rows



	spans	text
0	[8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,...	Another violent and aggressive immigrant killi...
1	[33, 34, 35, 36, 37, 38, 39]	I am 56 years old, I am not your fucking junio...
2	[0, 1, 2, 3]	Damn, a whole family. Sad indeed.
3	[7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]	What a knucklehead. How can anyone not know th...
4	[32, 33, 34, 35, 36, 37, 38]	"who do you think should do the killing?"\n\nA...

Test set sample rows

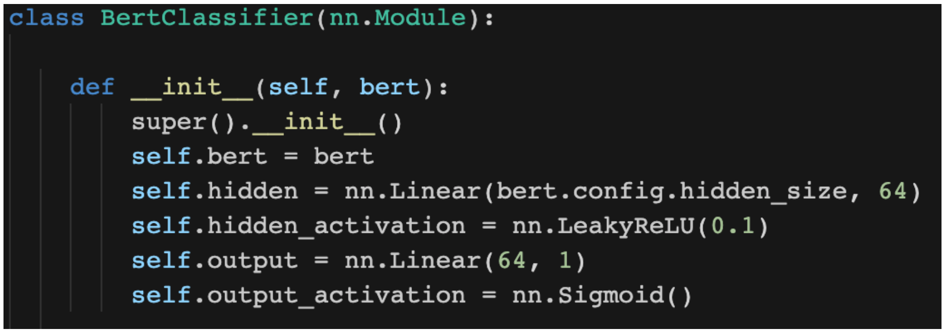


	spans	text
0	[84, 85, 86, 87, 88, 89, 90, 91, 133, 134, 135...	That's right. They are not normal. And I am st...
1	[81, 82, 83, 84, 85, 86]	"Watch people die from taking away their healt...
2	[]	tens years ago i contacted the PDR and suggest...
3	[]	The parallels between the ANC and the Sicilian...
4	[]	Intel Community: 'How can we work for a Presid...

Maximum length of a toxic span in the training set is 336 where as the maximum length of the toxic span in the test set is 291

Model parameters

The model has been trained over 3 epochs on binary cross entropy loss function and Adam optimizer. The model is being trained using Google collab on a GPU runtime.



```
class BertClassifier(nn.Module):

    def __init__(self, bert):
        super().__init__()
        self.bert = bert
        self.hidden = nn.Linear(bert.config.hidden_size, 64)
        self.hidden_activation = nn.LeakyReLU(0.1)
        self.output = nn.Linear(64, 1)
        self.output_activation = nn.Sigmoid()
```

Results

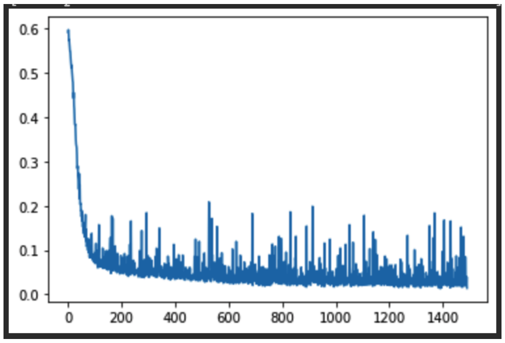
The model is evaluated based on the F1 score and the best score recorded is 0.6732. And the lowest loss recorded was 0.014272 on the training set.

F1 of the model is calculated as below

$$F_1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times |A \cap G|}{|A| + |G|}.$$

The score of the entire dataset is an average of the score of all sentences.

Loss curve



Conclusion

The toxic span detection task is a challenging problem in natural language processing, and this model is just a baseline approach. It is possible to improve the performance of the model by using more complex architecture and/or fine-tuning pretrained BERT models. Additionally, Conditional Random Field (CRF) models have been shown to achieve better performance on this task.

Note : F1 scores, false positive rates, true positive rates, losses and span predictions are recorded in a pickle file