MINI PROJECT REPORT

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Shared Task on Toxic Span Identification

BY :

1)Shresth Bharadia (19BCS100)
2)Sourav Bhagat (19BCS103)
3)Sutrave Krishna Sambhaji (19BCS105)

SUPERVISED BY : Dr. Sunil Saumya

ACKNOWLEDGEMENT

We would like to express our profound gratitude to **Dr. Sunil Saumya** (Assistant Professor, Computer Science and Engineering department) for his contributions to the completion of our project titled **TOXIC SPAN IDENTIFICATION**. We would also like to express our special thanks to **Mr. Shankar Biradar** (PhD Scholar, Computer Science and Engineering department) for his time and efforts provided throughout the semester. Their useful advice and suggestions were really helpful to us during the project's completion. In this aspect, we are eternally grateful to them. This project helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to them.

Shresth Bharadia (19BCS100)
Sourav Bhagat (19BCS103)
Sutrave Krishna Sambhaji (19BCS105)

1. INTRODUCTION

Detecting toxic posts on social network sites is a crucial task for social media moderators in order to keep a clean and friendly space for online discussion. To identify whether a comment or post is toxic or not, social network administrators often read the whole comment or post. However, with a large number of lengthy posts, the administrators need assistance to locate toxic words in each post to decide whether a post is toxic or non-toxic instead of reading the whole post. For example let's take a sentence " **Damn**, a whole family. Sad Indeed".In this sentence the word damn is **toxic** as shown in figure 1.

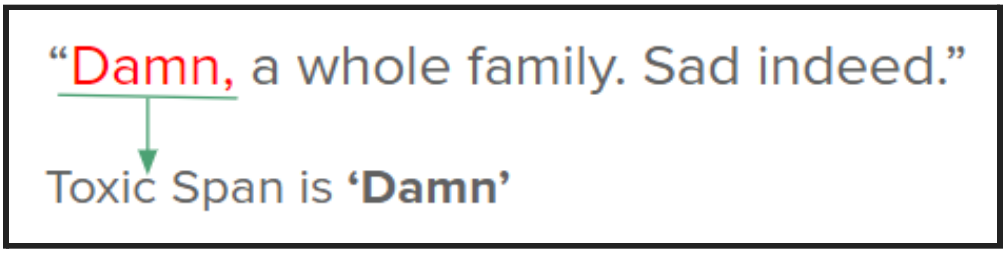

Figure1 : Damn is a toxic span in the given phrase.

The SemEval-2021 Task 5[1] provides a valuable dataset called Toxic Spans Detection dataset in order to train the model for detecting toxic words in lengthy posts. Based on the dataset from the shared task, we implement the machine learning model for detecting toxic words posts. Our model includes: the BiLSTM-CRF models with and without dropout layer, BiLSTM-Sigmoid for detecting the toxic spans in the post. Before training the model, we pre-process texts in posts and encode them by the GloVe word embedding. Our model achieves maximum accuracy of 60.03% on the test set provided by the task organizers.In addition, many shared tasks about hate speech and abusive languages are organized, such as the SemEval-2021 Task.

2. MOTIVATION

Moderation is crucial to promoting healthy online discussions. Although several toxicity (abusive language) detection datasets and models have been released, most of them classify whole comments or documents, and do not identify the spans that make a text toxic. But highlighting such toxic spans can assist human moderators (e.g., news portals moderators) who often deal with lengthy

comments, and who prefer attribution instead of just a system-generated unexplained toxicity score per post. The evaluation of systems that could accurately locate toxic spans within a text is thus a crucial step towards successful semi-automated moderation.

3. RELATED WORK

Many papers are written for toxic speech detection problems. They consist of flat label and hierarchical label datasets. The flat label datasets only classify one label for each comment in the dataset (e.g., hate, offensive, clean), while hierarchical datasets can classify multiple aspects of the comment (e.g., hate about racism, hate about sexual oriented, hate about religion, and hate about disability).

In addition, many shared tasks about hate speech and abusive languages are organized, such as the SemEval-2021 Task.

For the Toxic Spans Detection task, we adapt the mechanism from Sequence tagging and Name entities Recognition for detecting toxic words from posts.

4. DATASET

The dataset is provided from the SemEval-2021 Task 5: Toxic Spans Detection[2]. It includes the training and the test sets. Both of them consist of two parts: the content of posts and the spans denoting the toxic words in the posts as you can see in Table 1.

	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...

Table 1 : This table represents the top five samples of our dataset.

From table1 you can see that our dataset contains multiple spans of toxic words which are shown in the "spans" column which contains indices of character of a word, a phrase, or a sentence that is toxic in the text.

We analyzed that the Training dataset consists of 7939 rows. In which 4438 spans were unique in the text and the maximum frequency of spans in text was 485. The Test dataset consists of 2000 rows. In which 1034 spans were unique in the text. The maximum frequency of spans in text was 394.

5. Methodology

5.1. Data Cleaning and Data Preparation

With the given dataset from the SemEval-2021 Task 5 about Toxic Spans Detection, we firstly transform spans into a set of words. Then, we pre-process the posts as follows:

(1) Segmenting the posts by the TweetTokenizer from nltk2

(2) Changing texts to lowercase.

5.2. Data Preprocessing and Feature Extraction

We use the glove.twitter.27b.25d word embedding3 to construct the dictionary and encode the text of posts. Posts are encoded by the dictionary of the word embedding. The < UNK > tokens are added if a word in posts is not found in the dictionary. To make sure all vectors are the same length, we add the < PAD > token. Then, we set the maximum length of vectors

equal to 128. Spans are transformed into a one-hot vector corresponding to each word in posts where toxic words are denoted as 1 and others are denoted as 0. Table 2 illustrates an example of encoding data in our system.

	Original	**Transformed**
Text	I only use the word haole when **stupidity** and **arrogance** is involved and not all the time. Excluding the POTUS of course.	['i', 'only', 'use', 'the', 'word', 'haole', ..] **Vector:** [12, 216, 718, 15, 894,..]
Spans	[31, 32, 33, 34, 35, 36, 37, 38, 39, 45, 46, 47, 48, 49, 50, 51, 52, 53]	['i', 'only', 'use', 'the', 'word', 'haole', 'when', **'stupidity'**, 'and', **'arrogance'**, 'is', ...] **Vector:** [0, 0, 0, 0, 0, 0, 0, **1**, 0, **1**, 0, 0 ...]

Table 2 : Example of encoding data into vectors.

5.3. Methods Used

5.3.1 BiLSTM-CRF with or without dropout layer:

BiLSTM-CRF with or without dropout layer is a deep neural model used for Named-entity recognition tasks. We implement this model for the task of detecting toxic words in documents. The model includes three main layers:

(1) The word representation layer uses embedding matrix from the GloVe word embedding

(2) The BiLSTM layer for sequence labeling

(3) The Conditional Random Field (CRF) layer to control the probability of output labels.

The output is a binary vector, in which each value determines whether a word is toxic or non-toxic. The architecture of BiLSTM-CRF is described in Figure 2 and Figure 3.

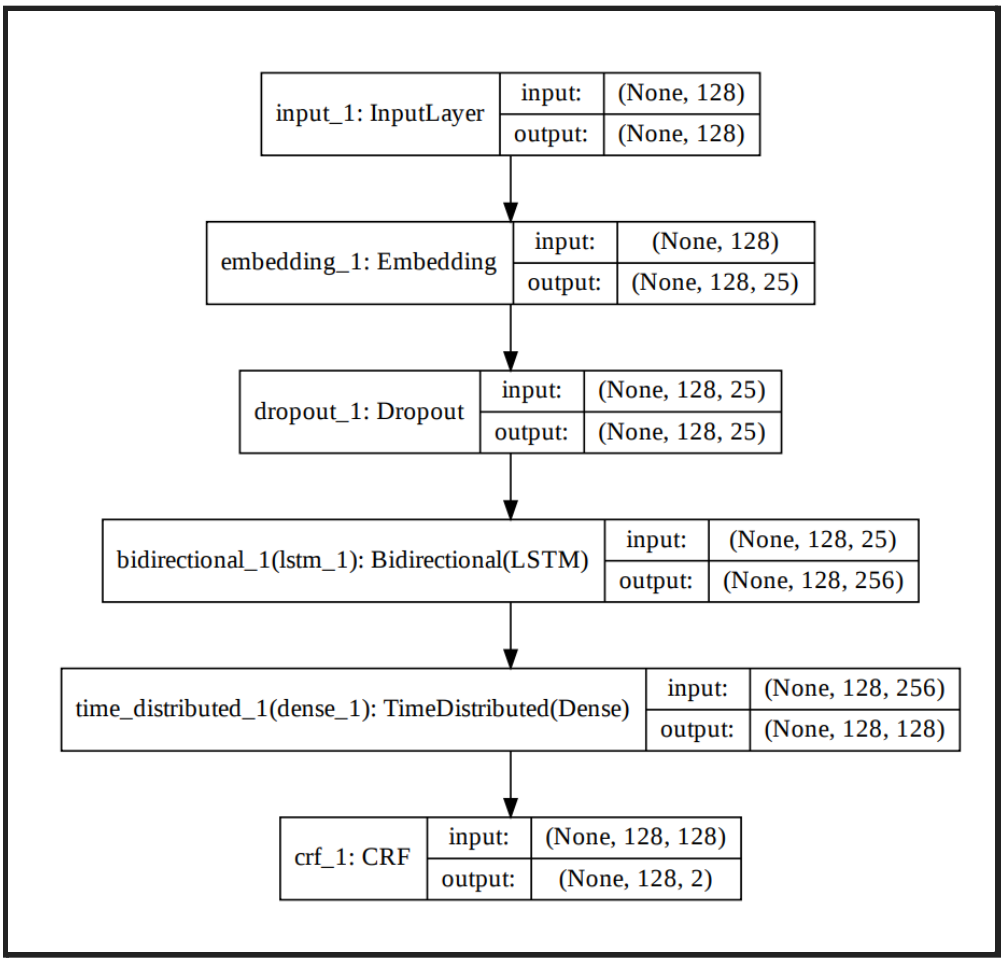

Figure 2: Architecture of BiLSTM-CRF with Dropout

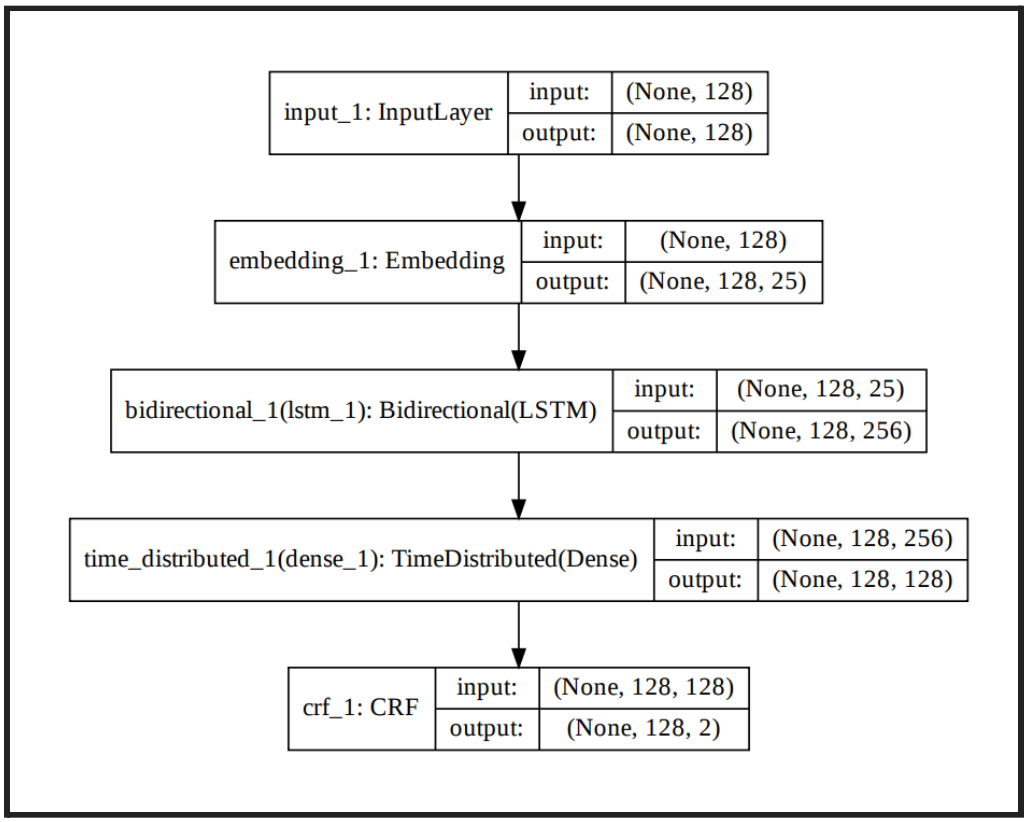

Figure 3: Architecture of BiLSTM-CRF with Dropout

5.3.2 BiLSTM-Sigmoid with or without dropout layer:

BiLSTM-Sigmoid with or without dropout layer is a deep neural model used for Named-entity recognition tasks. We implement this model for the task of detecting toxic words in documents. The model includes three main layers:

(1) The word representation layer uses embedding matrix from the GloVe word embedding

(2) The BiLSTM layer for sequence labeling

(3) The sigmoid function to control the probability of output labels.

The output is a binary vector, in which each value determines whether a word is toxic or non-toxic. The architecture of BiLSTM-Sigmoid is described in Figure 4 and Figure 5.

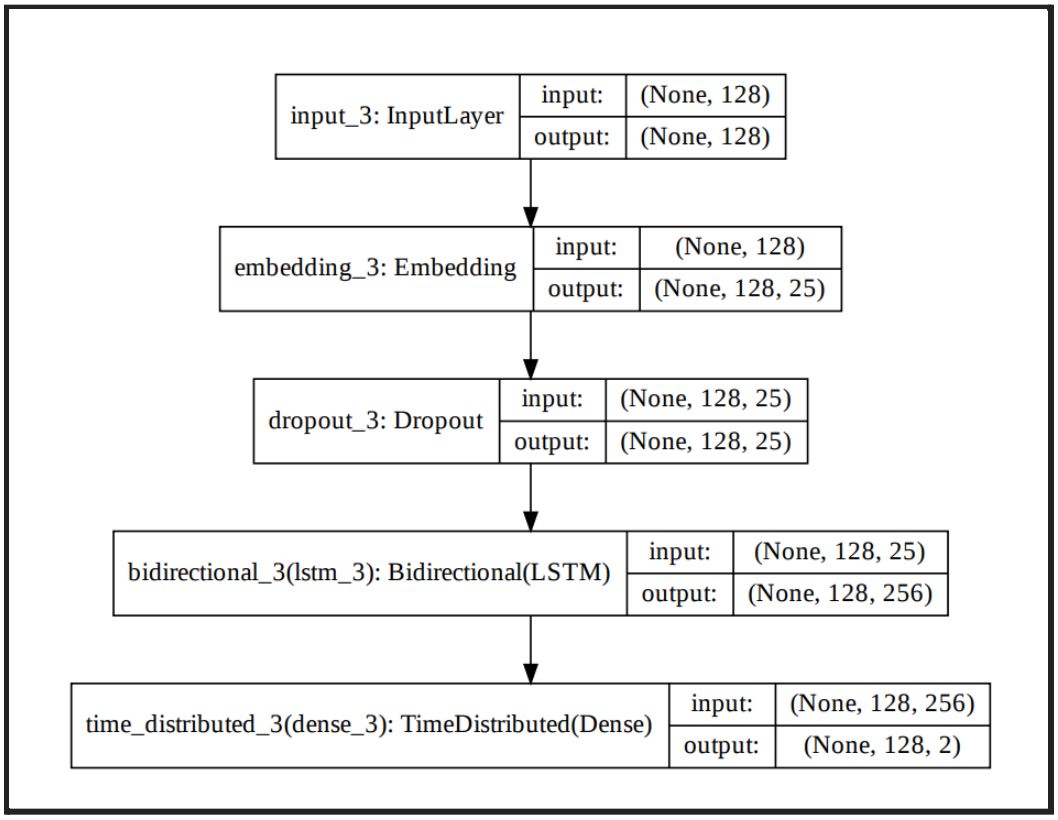

Figure 4 : Architecture of BiLSTM-Sigmoid with Dropout

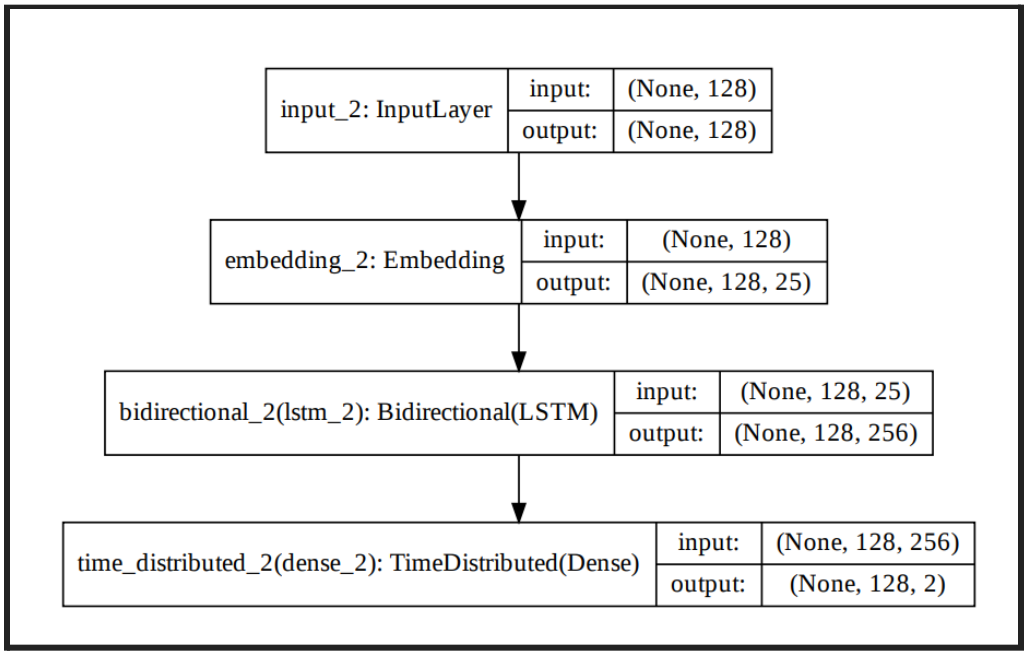

Figure 5 : Architecture of BiLSTM-Sigmoid without Dropout

5.4. Flow Diagram

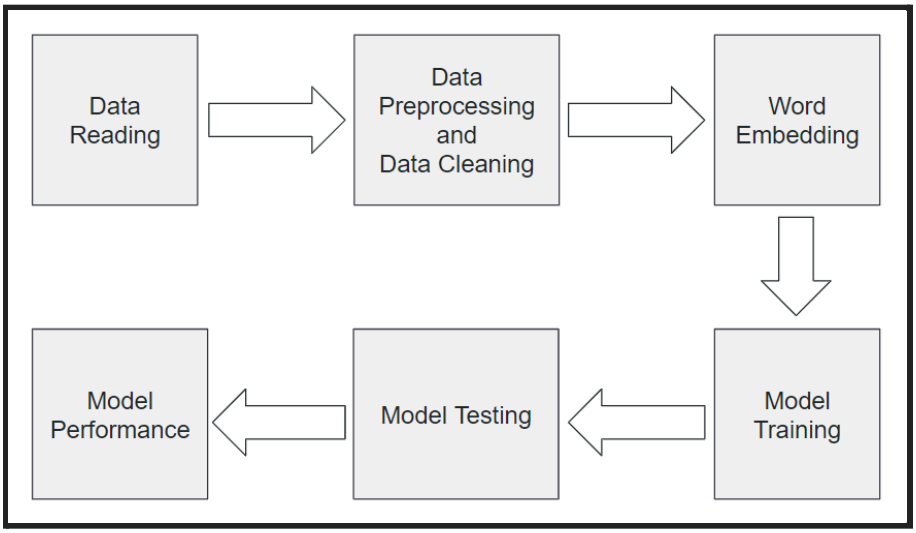

Figure 6 : The Flow Diagram

We first read the data and then did the data cleaning which includes removal of duplicate values and null values. Then we did data preprocessing which includes tokenization of the data and then spans are transformed into a one-hot vector corresponding to each word in posts where toxic words are denoted as 1 and others are denoted as 0.We use the glove.twitter.27b.25d word embedding3 to construct the dictionary and encode the text of posts. Posts are encoded by the dictionary of the word embedding. The < UNK > tokens are added if a word in posts is not found in the dictionary. To make sure all vectors are the same length, we add the < P AD > token. Then, we set the maximum length of vectors equal to 128.Then we used the Bi-LSTM CRF and Bi-LSTM Sigmoid model with and without dropout layer respectively for training the model. Then we used the F1 score for testing the model.

6. Results

6.1. Evaluation Matrix

The variant version of F1-score is used to evaluate the results of the competition. The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean.

$$F1 = \frac{2P*R}{P+R}$$

Where P is Precision and R is Recall of the Classifier.

6.2. Result for Model

Model	F1 Score
BiLSTM-CRF(Without dropout layer)	60.32%
BiLSTM-CRF(With dropout layer)	61.23%
BiLSTM-Sigmoid(Without dropout layer)	58.43%
BiLSTM - Sigmoid(with dropout layer)	59.06%

Table 3: Results obtained

According to Table 3, when BiLSTM-CRF without a dropout layer is used, the result by F1 score is 57.069%, when BiLSTM-CRF with a dropout layer is used, the result by F1 score is 60.03% , when BiLSTM-Sigmoid without dropout layer is used , the result by F1 Score is 58.23 and when BiLSTM-Sigmoid with dropout layer is used , the result by F1 Score is 59.06% as shown in Figure 7.

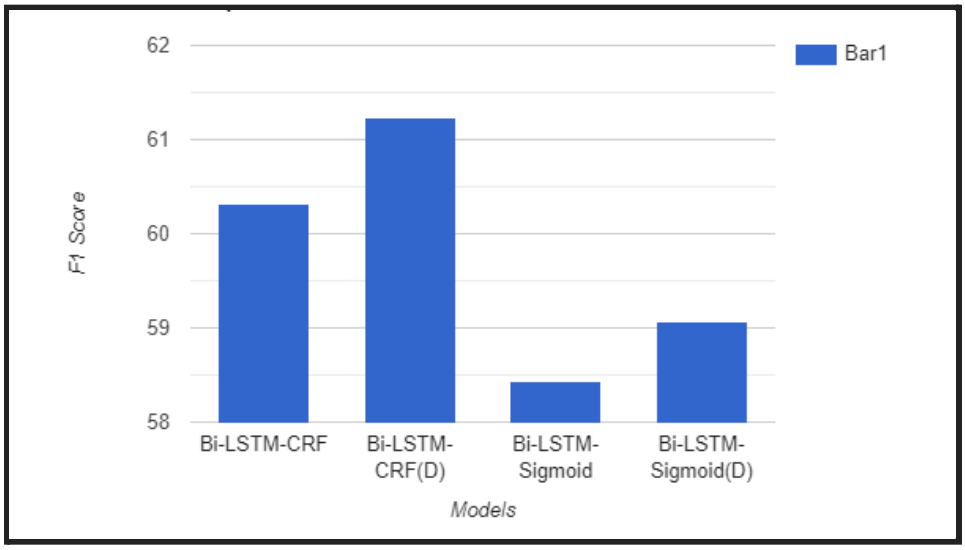

Figure 7: Comparison Between different models

7. Conclusion

We use the BiLSTM-CRF and BiLSTM-Sigmoid model with and without dropout layer respectively for detecting toxic words in the posts. Our BiLSTM-CRF with dropout model performs the best and has achieved the highest F1-Score of 61.23%. From the error analysis, we found that our model predicts well just for single-word spans and empty spans.

For further research, we can improve the performance of the detection model by applying the attention mechanism and using the character-level representation combined with word-level representation. Character-level models like CharBERT is a potential approach to increase the performance of toxic spans detection tasks.

8. References

1) SemEval 2021 Task 5: Toxic Spans Detection : https://competitions.codalab.org/competitions/25623#learn_the_details-overview (“Competition”)

2) https://github.com/ipavlopoulos/toxic_spans/tree/master/SemEval2021/data

3) https://aclanthology.org/2021.semeval-1.113/