Offensive Language Detection

1. **Problem Description:**

Nowadays, it became easy to express opinion and spread what we want through social media, but with the emergence and ease of this communication, we also have some problems such as cyberbullying and publishing offensive content on these platforms.

Therefore, it was necessary to find a solution to this problem such as tracking and detecting the offending posts automatically and deleting them.

1. **Model Design:**

For Modeling and Classifying we used a dataset that contain 13240 tweets.

Each instance it this dataset has three sub-tasks:

1. Sub-task A - Offensive language identification:

Has 2 values, (OFF) that means that tweet is offensive, or (NOT) that means that tweet is not Offensive.

1. Sub-task B - Automatic categorization of offense types:

Tweets that have value (OFF) in Sub-task A, have one of two values in Sub-task B; (TIN) Targeted Insult and Threats, or (UNT) Untargeted.

1. Sub-task C - Offense target identification:

Tweets that have (TIN) value in Sub-task B, have one of three values in Sub-task C; (IND) Individual, (GRP) Group, (OTH) Other.

FIRST:

We read data and split it into:

Tweet, subtask\_a, subtask\_b, subtask\_c, as shown in figure.

Graphical user interface, text, application, email

Description automatically generated

SECOND:

We preprocessed it by removing (patterns, emojis, stop words), then tokenizing, stemming, and lemmatizing tweets.

THIRD:

Vectorizing tokens using TF-IDF vectorizer to convert text data into numbers cause models deal only with numerical data.

FORTH:

For Classifying and Modeling, we make classify function that split data with ration 3:1 and use logistic regression model. The output of that function is the accuracy of training and testing data.

1. **Experimental Results:**

Results for testing data on model

* subtask\_a is 76.57%
* subtask\_b is 87.99%
* subtask\_c is 67.467%

1. **Model Performance**

Results for training data on model

* subtask\_a is 80.55%
* subtask\_b is 88.139%
* subtask\_c is 78.91%