Universidad Distrital Francisco José de Caldas

Systems Analysis: Toxic Comment Classification Challenge

Hugo Mojica Angarita 20232020034
Laura Paez Cifuentes 20232020055
Andrey Camilo Gonzales Caceres 20231020070

Professor: SIERRA VIRGUEZ CARLOS ANDRÉS

System analysis and Design
Systems Engineering
April 5, 2025

Selected Competition: Toxic Comment Classification Challenge

"The Conversation AI team, a research initiative founded by Jigsaw and Google (both a part of Alphabet), are working on tools to help improve online conversation. One area of focus is the study of negative online behaviors, like toxic comments (comments that are rude, disrespectful or otherwise likely to make someone leave a discussion). So far they've built a range of publicly available models served through the Perspective API, including toxicity. But the current models still make errors, and they don't allow users to select which types of toxicity they're interested in finding (e.g. some platforms may be fine with profanity, but not with other types of toxic content). In this competition, you're challenged to build a multi-headed model that's capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate better than Perspective's current models. You'll be using a dataset of comments from Wikipedia's talk page edits. Improvements to the current model will hopefully help online discussion become more productive and respectful." "

1. Key Elements of the Competition

The data sets given in the competition are designed to train, test and analyze toxicity detection models.

Key Files

- The dataset includes many comments that have been classified by human evaluators for toxic behavior.
- **Training set (`train.csv`):** It is a training set containing comments with their binary tags.
- **Test files:** This file contains unlabeled comments, the comments on which the models should predict toxicity. It is used to generate predictions that are then evaluated in the Kaggle Leaderboard.

Features

- **Toxicity labels:** For example, `severe_toxicity`, `obscene`, `identity_attack`, etc. These variables help to understand the multidimensionality of toxicity: not all toxic comments are the same.
- **Identity mentions:** Indicate whether the comment mentions features such as male, female, black, `sexual_orientation`, etc. This is essential to evaluate if the model is systematically penalizing certain groups by mentioning their identity without insulting them.
- **Metadata:** Such as dates, identifiers, and annotator counters.

Target Variables

- **toxicity_annotator_count:** Number of annotators who evaluated the toxicity.
- **identity_annotator_count:** Number of annotators who identified identity mentions.
- **target:** Main target variable as it indicates the fraction of annotators who considered a comment to be toxic. Its value is a number between 0 and 1, for example, if `target` = 0.75 means that 75% of the annotators consider the comment toxic.

Additional Considerations

- Kaggle provided special metrics to measure this behavior: Subgroup AUC, BPSN AUC (Background Positive, Subgroup Negative), BNSP AUC (Background Negative, Subgroup Positive).

2. Relationship Mapping

2.1. System Elements and Their Relationships

- **Input (comments: comment_text):** Input element of string type. It contains the comments to be analyzed from the users, varying from neutral phrases to insults, threats, humor, and affirmations. It is important to consider comments that seem toxic but may be sarcastic, as well as the context of each one.
- **Identity annotations:** Variables that indicate a mention. They have continuous values from 0.0 to 1.0, where the number represents the fraction of annotators who think that identity has been explicitly or implicitly mentioned. **Examples of identity:**
  - Gender: male, female, transgender, other_gender
  - Race/ethnicity: black, white, latino, asian, other_race_or_ethnicity
  - Religion: christian, muslim, jewish, buddhist, atheist, other_religion
  - Sexual orientation: heterosexual, homosexual_gay_or_lesbian, bisexual, other_sexual_orientati
  - Health conditions/disability: physical_disability, psychiatric_or_mental_illness
- **Toxicity Tags:** In the dataset, the columns perceive different types of toxicity in a comment:
  - `toxicity`: general level of toxicity.
  - `severe_toxicity`: extreme toxicity and intense verbal violence.
  - `obscene`: vulgar or inappropriate language.
  - `identity_attack`: attacks directed at identity groups.
  - `insult`: personal or collective insults.
  - `threat`: explicit or implicit threats.
  - `sexual_explicit`: explicitly sexual content.

- **Target Variable (target):** It is a variable between 0.0 and 1.0 that represents the level of toxicity that the annotators considered the comment to have. This variable is the one that the model must predict.

2.2. System Information Flow

The flow of information in this context is useful to understand how information flows within the system, such as what data is generated, its destination, who uses it, and how it changes. This helps to visualize the relationships more clearly.

1. **Step 1: Text input.** The user's comment is received as input into the system.

2. **Step 2: Processing.** The input is cleaned, leaving only a string without symbols, capital letters, or extra punctuation. In addition to tokenizing it (processing the natural language to make it easier for machines to understand), using some model.

3. **Step 3: Identity mentions.** The identity mentions of the comment are analyzed, and sensitive mentions are identified. This can be achieved by using rules or an additional model.

4. **Step 4: Classification of the comment.** The model must process the comment and make a toxicity prediction to store it in the target variable. As an addition, predictions for `severe_toxicity`, `identity_attack`, etc., can be added.

5. **Step 5: System output.** Returns the toxicity probability, the target variable.

2.3. System Constraints

The competition provides certain rules, such as the use of public data or data provided by Kaggle. Regarding RAM, the code must run within a limited time. Additionally, the data provided cannot be modified.

It is important to emphasize some ethical rules, such as avoiding training the model to develop discrimination towards identity groups, as the model must minimize these behaviors.

2.4. System Boundaries

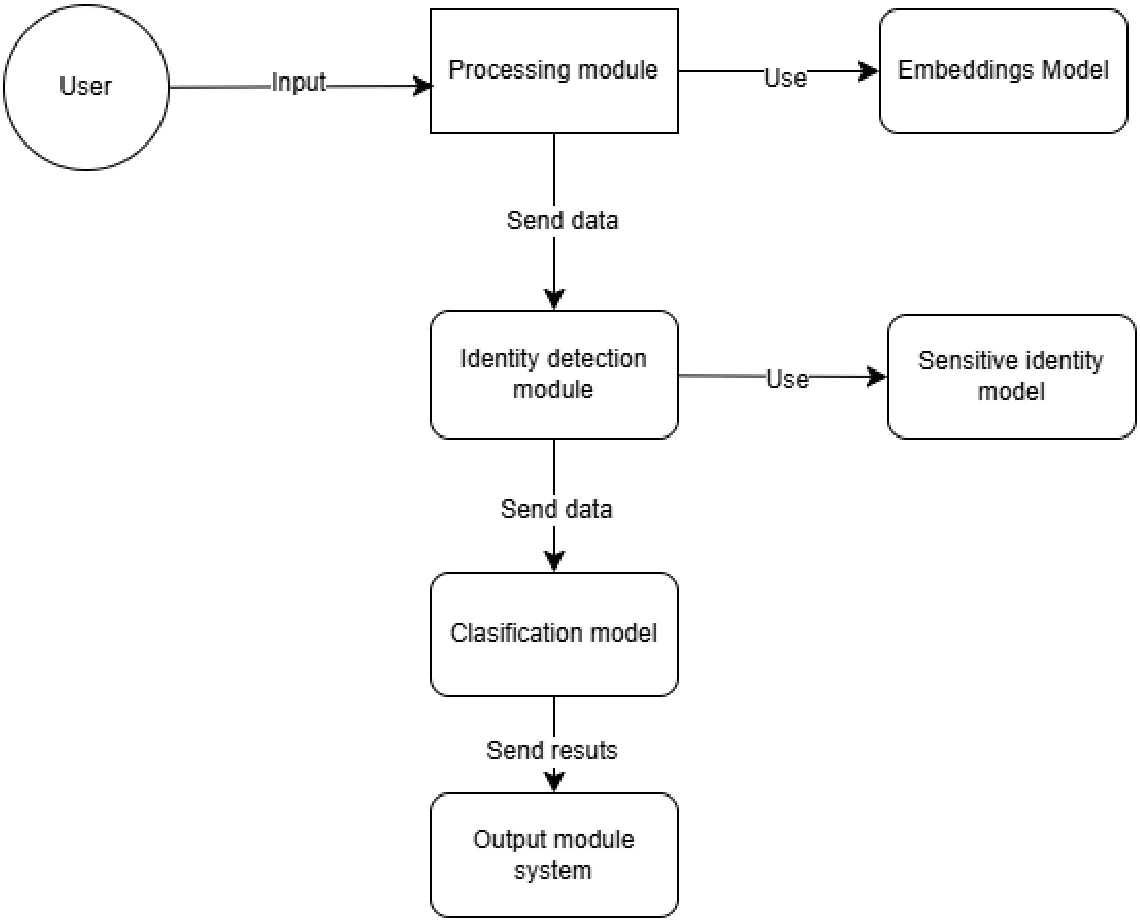
Defining the boundaries helps us to identify what part of the system is and what is outside its control.

The elements that are within the system boundaries are: input data, text preparation process, classification model, evaluation metrics, model validation, and output.

The elements that are outside the boundaries of the system are: the moral or legal interpretation that is outside the perception contained in the data, the generation of real-time responses, and the perception of the language of all the different social identities.

In conclusion, the system helps us to mitigate toxicity in a moderation ecosystem; however, it does not fully address the social impact and ensure structured justice.

2.5. Relationship Mapping Graph



3. Systems Engineering Perspective – Systemic Analysis

This section aims to study the ”Jigsaw Unintended Bias in Toxicity Classification” competition as a complex system that can be analyzed using the principles of systems engineering. This involves breaking down its components, understanding how they interact, their limits, objectives, inputs and outputs, as well as the social, technical, and ethical context of their environment, life cycle, and other aspects that have been addressed in classes throughout the semester.

3.1 System Definition

From this perspective, the main system we analyze is a text classification model designed to predict the toxicity of natural language comments. However, it is not just about detecting toxicity, but also ensuring fairness: the system must minimize biases that affect certain social groups, especially those that have been historically vulnerable.

This system is composed of several elements that collaborate to fulfill a primary function: helping to moderate online content, reducing the harm caused by toxic speech while ensuring that it does not unduly punish those belonging to minorities or mentioning sensitive identities.

3.2 System Requirements

Functional Requirements (What the system should do):

- Detect toxicity: Predict the toxicity of a comment (value between 0 and 1).
- Recognize identities: Determine whether the text mentions terms associated with gender, race, religion, or other social groups.
- Handle multiple annotations: Integrate opinions from multiple human annotators to estimate the collective perception of toxicity.
- Evaluate fairness: It should not only perform well in general, but also with comments related to minorities, avoiding systematic bias.
- Offer comprehensive metrics: Include overall AUC, Subgroup AUC, BPSN AUC, and BNSP AUC to evaluate performance from various angles.

Non-Functional Requirements (How it should work):

- Computational efficiency: The system must be able to handle thousands of comments in a reasonable amount of time.
- Scalability: It must be able to adapt to larger volumes without the need to redesign the architecture.
- Transparency: Its decisions must be understandable to human users and auditors.
- Reproducibility: The results must be exactly replicable under Kaggle's conditions.
- Resource constraints: It must comply with the time and memory limits imposed by the competition platform.

3.3 System Components

We can divide the system into three main blocks:

Input:

- Free-text comments (`comment_text` field).
- Binary columns that identify mentions of identities (male, female, Christian, black, etc.).
- Annotations from multiple people indicating their perception of toxicity (`toxicity_annotator_cou` `target`, etc.).

Processing:

- Preprocessing: Text cleaning, special character removal, tokenization, lemmatization, etc.
- Feature extraction: Text representation using embeddings, TF-IDF, bag-of-words, or more advanced models such as BERT.
- Modeling: Application of classification models (from logistic regression to deep neural networks).
- Metric calculation: Evaluation of model performance from a general and subgroup perspective.

Output:

- Continuous predictions between 0 and 1 indicating the probability that a comment is toxic.
- Analytical reports by subgroup, visualizations, and bias metrics.

3.4 System Environment

Target users:

Social platform moderators, AI ethics researchers, developers of automatic moderation systems.

External factors:

- Linguistic changes (new forms of insult, idioms).
- Cultural differences (what is offensive in one region may not be offensive in another).
- Legislative changes regarding freedom of expression or content moderation.
- Data sensitivity: Comments contain explicit references to identity, religion, or race, so ethical handling is crucial.

3.5 System Restrictions

Technical:

- Limited resources when running the code (memory and time) on Kaggle kernels.
- The use of any additional data not provided by the organizers is prohibited.

Evaluative:

- The metrics used by the competition prioritize not only accuracy but also the fairness of the model across different subgroups.

Human:

- Labels come from humans with different backgrounds and criteria. The perception of toxicity is not universal.

3.6 System Lifecycle

Phase: What this competition entails

- **Design:** Objectives are set (detecting toxicity and reducing bias). Inputs, outputs, and constraints are defined.
- **Development:** The complete pipeline is built: data cleaning, model selection, and metrics definition.
- **Validation:** Models are tested with different configurations, measuring overall performance and performance by subgroup.
- **Implementation:** (Although not a direct part of the challenge, in a real-life scenario, it would be integrated into an automated moderation system.)
- **Maintenance:** (Hypothetical) The model would be updated with new data, reviewed for biases, and adapted to cultural changes.

3.7 Ethical and Social Considerations

These types of systems do not operate in a vacuum: they analyze human language, which is loaded with culture, emotions, and subjectivity. Therefore, ethical analysis is key.

- **Risk of algorithmic bias:** A poorly trained model may penalize certain groups more heavily, even if their language is not more toxic.
- **False positives:** A legitimate comment can be removed simply for mentioning an identity.
- **False negatives:** Subtle offensive comments (such as sarcasm or double entendres) may go unnoticed.
- **Negative feedback:** If the system reinforces historical biases, it may amplify harm rather than mitigate it.

In short, the system must try to understand not only what is being said, but also how and who is saying it, always with a responsible and empathetic approach.

4. Sensitivity Analysis

This analysis is extremely important when training machine learning models because it helps us understand how much each decision we make (for example, how we prepare the text, what parameters we use, etc.) affects the model's final result. In the case of this Jigsaw competition, where we seek to detect toxic comments without indulging in bias, this is key.

4.1. What affects the model

There are several decisions that can make our model perform better or worse. Here are some of the main ones:

Text representation:

The way we convert the text into something the computer can understand is critical. Some common techniques we tested were:

- TF-IDF: basically gives more weight to words that appear frequently in a comment but are not as common in all texts.
- Word Embeddings (such as Word2Vec or GloVe): convert words into vectors that ”understand” the meaning of words based on how they are used.
- BERT: This is more advanced because it takes into account the context of each word within the sentence.

Depending on the technique you use, the model will understand things like sarcasm or double entendres better or worse.

Model hyperparameters:

These are like the ”controls” we use to fine-tune how the model learns. Some examples:

- Learning rate: how quickly the model adapts to errors.
- Batch size: how many comments the model sees before updating.
- Epochs: how many times it goes through the entire dataset.
- Regularization: so that the model doesn't retain the exact memory of the data and can generalize.

If you don't choose these correctly, your model may fail due to over- or under-learning.

Text preprocessing:

Before training, the text must be neat. Some things we did were:

- Tokenization: splitting the comment into words or phrases.
- Lemmatization or stemming: Reducing words to their basic form (for example, “insulting” → “insult”).
- Removing common words that don't contribute much (“and,” “the,” “of”).

Doing this well improves the quality of the data we use for training, and this is noticeable in the final result.

Strategies to reduce bias:

Since the goal is to prevent the model from being unfair to certain groups, there are techniques that help:

- Reweighting: Giving more weight to certain examples to maintain balance between different groups (for example, not all comments with the word "Muslim" are labeled as toxic).
- Threshold adjustment: Changing the number above which we consider something toxic.
- Adversarial debiasing: Training the model to ignore people's identities when making predictions.

Each strategy works differently depending on the model and the data, so it's worth testing.

4.2. How we measure all of this

There are tools that allow us to see the impact of these parameters and decisions. Some of them are:

Cross-validation:

We divide the dataset into parts and train using different combinations. This way, we ensure that the model doesn't rely solely on one group of data and becomes more general.

Learning curves:

We create graphs to see how the model performs as it sees more data. This helps us understand if we are overtraining (overfitting) or if the model lacks the ability to learn (underfitting).

Feature importance:

This allows us to see which words or groups of words (n-grams) are most useful for the model to detect toxicity. This can also give us clues about bias (for example, if the word "gay" appears as very important for detecting toxicity... something is wrong).

Fairness metrics:

These help us see if the model treats all groups equally. Some metrics we use:

- Equal opportunity: that the true hit rate is equal between groups.
- Demographic parity: that positive predictions are distributed equally.
- Disparate impact: compares how much a disadvantaged group is affected compared to a privileged one.

With this, we can compare how fair a model is and whether our strategies to reduce bias are working.

Local sensitivity analysis:

Here, what we do is change a comment slightly (for example, change a word) and see if the model completely changes the prediction. If that happens, it's because the model isn't very stable or is too sensitive to specific words.

5. Chaos and complexity theory

This part sounds a bit more theoretical, but it makes sense when applied to how a machine learning model works. In essence, chaos theory says that in complex, nonlinear systems (like human language), a small change can have a huge effect. And ML models, especially in NLP tasks, are full of these situations.

5.1. Chaotic or Unpredictable Elements

Language Variability:

The way we speak is a mess (and a beautiful one at that). It can contain irony, sarcasm, puns, insults disguised as jokes, etc. The model often can't capture these subtleties. Furthermore, the dataset includes strong comments, and that adds a lot of ”noise.”

Subjectivity of Toxicity:

What may be offensive to one person may not be to another. In the dataset, toxicity is based on how many annotators marked a comment as toxic. This means that even from the labeling, there is some ambiguity.

Data Biases:

Training data may contain biases that reflect real prejudices. For example, if more comments with certain identities were labeled as toxic in the original data, the model may learn that as if it were correct and continue reproducing those biases.

Interaction between participants:

As this is a public competition, each person or team tries new things, and this means the environment is constantly changing. Ideas are shared, and this can cause the ”state of the art” to change from one week to the next. It's a dynamic system.

5.2. Random or feedback processes

Text generation:

The way toxic comments are generated is not predictable. It depends on the mood of the person writing it, the context, etc. In other words, it's not something we can easily model.

Annotation process:

Labeling data also has its random side. The person annotating it may be tired, distracted, or have their own biases. Even the same comment viewed by two people can receive different labels. This is why multiple annotators are sometimes used per comment.

Model Training:

The learning process of a model is not linear. Small changes in the data, parameters, or even the order in which examples are viewed can significantly change the outcome.

Competition Dynamics:

Because it's an open competition, one team's results and strategies can influence others. This creates a kind of feedback loop where everyone constantly improves and adapts.