5-2021

Applying Emotional Analysis for Automated Content Moderation

John Shelnutt

Applying Emotional Analysis for Automated Content Moderation

by

John Shelnutt

A thesis submitted in conformity with the requirements
for the degree of Honors in Computer Science

Undergraduate Department of Computer Science and Computer Engineering
University of Arkansas

Abstract

The purpose of this project is to explore the effectiveness of emotional analysis as a means to automatically moderate content or flag content for manual moderation in order to reduce the workload of human moderators in moderating toxic content online. In this context, toxic content is defined as content that features excessive negativity, rudeness, or malice. This often features offensive language or slurs. The work involved in this project included creating a simple website that imitates a social media or forum with a feed of user submitted text posts, implementing an emotional analysis algorithm from a word emotions dataset, designing a system to configure tolerance thresholds on a per-emotion basis, implementing the process of determining violations of incoming text posts using the configuration, and testing the effectiveness of the emotional analysis algorithm at determining toxic posts using a dataset of posts that have been manually reviewed for toxicity by a group of human moderators.

Contents

Chapter 1

Background

1.1 Introduction

Online forums provide a space for users to interact and share thoughts and ideas, often in the form of text posts. The freedom provided by these spaces online also invites unwanted content like arguing, threats, offensive language, harassment, and hate speech. [6]. The responsibility of removing this content falls on both human moderators and scripts/bots that attempt to moderate as effectively as humans but on a much larger scale. Online moderation is important because it maintains the quality of discussions worldwide and covers millions of user submitted content. In 2018 alone, the popular website Reddit amassed over 80 million user submissions. [6]. In 2017, the site Pinterest reported having only 11 full-time moderators compared to the website's 200 million users. [8] Popular methods to automatically moderate content include keyword detection of profanity and simple pattern matching using regular expressions. [6]. To date, no widespread moderation solutions exist that leverage emotional analysis [5] to assist moderators online in identifying and moderating content in violation of site guidelines.

The purpose of this project is to explore the effectiveness of emotional analysis as a means to automatically moderate content or flag content for manual moderation in order to reduce the workload of human moderators in moderating toxic content. In this context, toxic content is defined as content that features excessive negativity, rudeness, or malice. This often features offensive language or slurs. The work involved in this project included creating a simple website that imitates a social media or forum with a feed of user submitted text posts, implementing an emotional analysis algorithm from a word emotions dataset, designing a system to configure tolerance thresholds on a per-emotion basis, implementing the process of determining violations of incoming text posts using the configuration, and testing the effectiveness of the

emotional analysis algorithm at determining toxic posts using a dataset of posts that have been manually reviewed for toxicity by a group of human moderators.

1.2 Emotional Analysis vs. Sentiment Analysis

Both sentiment [9] and emotional [5] analysis are approaches to determining the overall positivity and negativity from a body of text. These analysis methods can vary significantly in implementation. Typically, some form of tokenization and classification is performed. Analysis of either kind could involve the use of machine learning to determine parts of speech and connotations of words or groupings of words. Mackey's research notes that many state-of-the-art applications leverage unsupervised machine learning to extract emotion data using pre-trained word embeddings. [7] [1] For this project, emotional analysis was performed by tokenizing inputs and mapping individual words to a vector of emotional scores using a word emotions dataset. More details about this can be found in section 3.2.

Sentiment analysis is a method of analysis that converts textual input into a scalar value. A positive number indicates a positive sentiment, while a negative number indicates a negative sentiment. A value of 0 indicates a neutral sentiment.

Emotional Analysis, in contrast, converts the text into a vector containing scores for individual emotions. Where sentiment analysis would result in a negative number, emotional analysis provides the additional insight of which particular negative emotions are exhibited in the text—anger, fear, disgust, joy, sadness, and surprise. These 6 emotions were identified by psychologists in the late 20^{th} century as being *universal*, or applying to all humans irrespective of culture or upbringing. [4] [3] This project uses a dataset that provides additional emotions such as "trust". The full list can be seen in table 3.1.

1.3 Moderation as a two stage process

The moderation of textual content online such as text posts and comments can be viewed as a two-stage process. The first stage is automatic filtering which can act as a firewall to moderate content before the content is posted for others to see. The second stage is where manual moderation happens. For most large sites that feature user submitted content, this is in the form of a team of human moderators that actively browse user submitted content and perform deletions and bans. Typically the first stage or automatic moderation occurs instantly at the time of posting, while the second stage occurs after a period of time when the human moderators are active

and reviewing recent posts.

1.3.1 Automatic moderation

The most common use of automatic moderation is to perform simple scanning of text to preemptively moderate posts that definitively violate site rules. Such posts contain vulgar content like offensive language and slurs. The scope of automatic moderation has been intentionally limited by sites to avoid false positives, which results in a poor user experience. First stage moderation is effective at filtering blatant violations and extreme toxicity, which can easily be detected using keyword-checking. As a consequence of this, the work of moderating the more subtle yet still toxic posts falls under the second stage—human moderation.

1.3.2 Human moderation

Human moderation is a delayed moderation stage in which real people review incoming submissions. This kind of moderation naturally has a delay, as most active websites have a stream of incoming posts whose throughput is more than the moderation throughput that even a team of moderators can sustain. However, human moderation is more effective at catching toxic posts that don't contain explicit language. Additionally, having a team of human moderators who read all user submitted content is a time-consuming and expensive process that often lets toxic posts go unmoderated for a while or go unnoticed altogether. This is a reasonable trade-off for most online forums due to the nuance that human moderators have in analyzing user submissions.

1.4 Goals

The emotional analysis work in this project aims to assist in the second stage of moderation that requires human intervention. By automatically performing emotional analysis on all incoming user submitted content and leveraging a predefined configuration that defines tolerances for negative emotions, posts that do not get filtered out by the profanity filters in the first stage could be automatically moderated or flagged for review in the second stage. **This would greatly reduce the workload on human moderators** by limiting the number of posts that require human review from all user submitted content that gets past the first stage to only posts that get flagged for review by the emotional analysis moderation tool.

Chapter 2

Literature Review/Related Work

The research conducted by Jhaver et al [6] studies the human-computer interaction in the popular online forum and discussion site Reddit. Jhaver et al outline the functionality and effectiveness of a popular moderation tool called "AutoModerator". Their work details the online moderation landscape and common techniques like pattern matching using regular expressions. Strapparava and Mihalcea's research details several approaches to the implementation of an emotional analysis algorithm. [11]. They cover both knowledge-based and corpus-based emotion annotation. The knowledge-based approach more closely aligns with the approach in this project, which utilizes a dataset that maps words to their affective emotions. Their baseline algorithm annotates emotions in a given text using the presence of words in the emotional lexicon, on which the emotional analysis algorithm in this project was modelled from. Barnes, Klinger, and Walde address the cutting-edge of learning algorithms and the efforts to improve upon dictionary techniques using supervised machine learning over lexical features like word embeddings. [1] These approaches typically involve pre-trained word embeddings to project words into a vector space of emotions as a function of context words. [7] The focus of this project takes inspiration from the work of University of Arkansas PhD candidate Andrew Mackey. His guidance in the work of this project involved supplying the datasets used in the emotional analysis algorithm and in testing the effectiveness of the project at classifying toxic input text. Mackey's ongoing work involves the application of emotional analysis to detect "fake news". [7]

Chapter 3

Implementation

3.1 Technology Stack

This project was written using Node.js for the back-end, which is a JavaScript runtime based on the Google Chrome V8 engine. The API was created using Express, a popular Node.js package for defining custom HTTP routes and behavior. In order to serve a dynamic feed of posts, the templating engine EJS was used. Persistent data was stored using a PostgreSQL Database and interfaced with through the popular ORM Sequelize. The user interface was styled using Bootstrap, which provides out-of-the-box styling for HTML elements.

3.2 Emotional Analysis Algorithm

The emotional analysis algorithm uses an emotion lexicon, the NRC Word-Emotion Association Lexicon (EmoLex), manually produced by Mohammad and Turney. [10] The lexicon contains 14,182 English words, each tagged with 1 or 0 for each of the eight emotions and two connotations as shown in table 3.1

Each line in the lexicon consists of three tab-separated values: an English word,

Table 3.1: Dataset categories

emotion	**connotation**
anger	positive
anticipation	negative
disgust	
fear	
joy	
sadness	
surprise	
trust	

one of the categories from table 3.1, and a 0 or 1, indicating if the word exhibits that emotion or connotation. See table 3.2 for an example of an entry in the dataset.

Table 3.2: Lexicon structure

...		
demise	anger	0
demise	anticipation	0
demise	disgust	0
demise	fear	1
demise	joy	0
demise	negative	1
demise	positive	0
demise	sadness	1
demise	surprise	0
demise	trust	0
...		

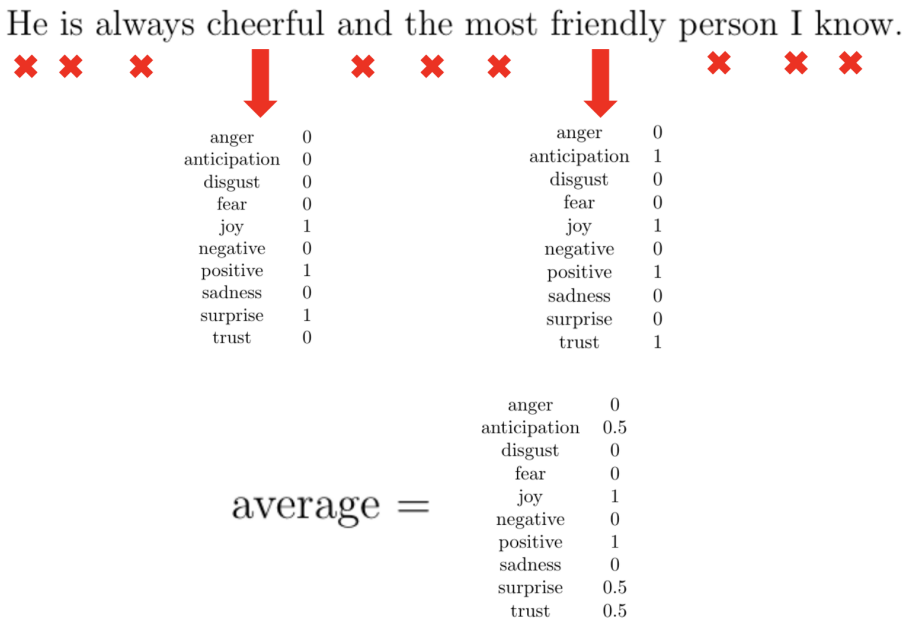
Using this emotion lexicon, the application starts by initializing a hash map so that emotions for a given word can be looked up in $O(1)$ time. This is performed on startup, and the hash map is reused for the remainder of the application's runtime. When the API receives a request to create a new post, the text of the post is then tokenized into an array of words. Each word is then mapped to its corresponding emotion scores using the hash map. If a word is not present in the hash map, it is discarded and not considered for the remainder of the calculation. Two different metrics are then computed—raw scores and percent scores.

3.2.1 Raw Scores

Raw scores for each category are computed by summing the score for that category (0 or 1) for each word that was found in the hash map. This total is then divided by the number of words that were found in the hashmap to get the average value for each emotion/connotation. The result is a vector containing a numerical value for each emotion and connotation that represents the average value for that emotion/connotation over the all of the words in the input text that were contained in the lexicon. Computing an average value over the input text prevents being biased against longer posts. Additionally, the total is divided by the number of words present in the lexicon rather than the actual total number of words in the post in order to prevent the result from being too lenient on input text containing a lot of words not present in the lexicon. These could be unrecognized words such as slang words or "filler" words that were not included in the lexicon (of, the, and, ..., etc.) Including these words in the total count would unfairly bring down the average, so unrecognized words are ignored in the calculations. Figure 3.1 shows the calculation of raw scores for a simple

input text. Note that the phrase "most friendly" is not scored more positively than just "friendly" on its own. Modifiers like "so", "very", "really", etc. do not amplify the scoring of the words they modify. Phrases like "not happy" will be misinterpreted as having a positive connotation, whereas "unhappy" will be correctly interpreted as negative. Every word is treated in isolation to the other words, which is one drawback to this algorithm.

Figure 3.1: Example calculation of raw scores



Note that the average is taken by dividing the sum of each emotion by the number of words that contain emotion data. The example shown in figure 3.1 is calculated by dividing the emotion sums by 2, since only 2 words have emotional data from the lexicon. If this sentence was reworded to contain more or less prepositions or otherwise neutral parts of speech, the computed score would remain the same.

3.2.2 Percent Scores

Percent scores are computed based on the raw scoring process mentioned above. The key difference is that the percent scores provide a way to determine the amounts of particular emotions and connotations relative to each other. For this calculation, "positive" and "negative" are considered connotations, and the rest are considered emotions. For both the connotations and the emotions from the raw scores, the values are summed and the percentage of one particular emotion or connotation is computed. Equation 3.1 below shows the calculation of the percent score for the

positive connotation.

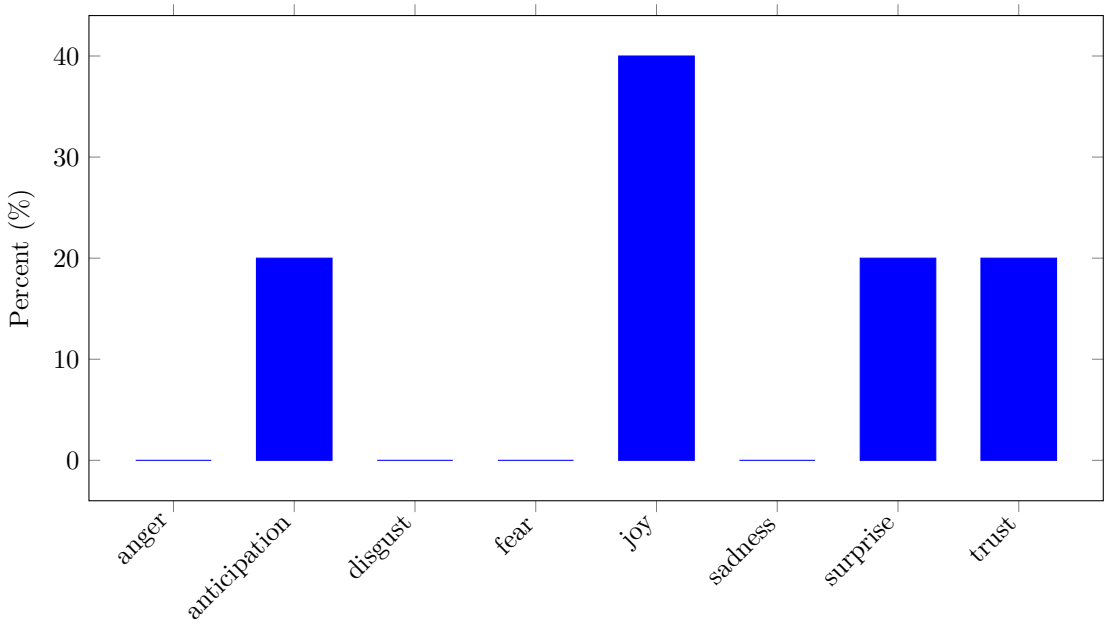
$$\text{positive}_{\text{percent score}} = 100 \times \frac{\text{positive}_{\text{raw score}}}{\text{positive}_{\text{raw score}} + \text{negative}_{\text{raw score}}} \tag{3.1}$$

Similarly for calculating the percent scores for an emotion, the sum of the raw scores for all emotions are used in the calculation to determine the relative amount of one emotion to the sum of all emotions displayed in the input text. Equation 3.2 shows the calculation performed with anger as an example.

$$\text{anger}_{\text{percent score}} = 100 \times \text{anger}_{\text{raw score}} \Big/ \left(\sum_{r \in R} r \right), \text{ where } R \text{ is the set of raw scores} \tag{3.2}$$

Figure 3.2 shows the percent scores of emotions when applied to the example input text from Figure 3.1.

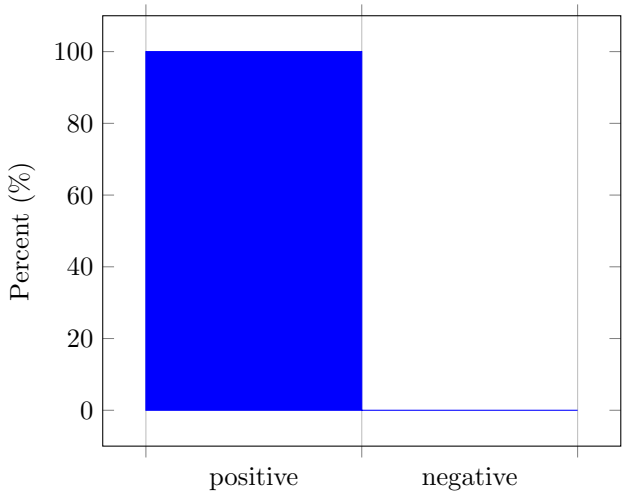
Figure 3.2: Percent scores applied to emotions from previous example.



By computing emotion scores as percents, it is now known that the emotion "joy" makes up almost half of the displayed emotions from the input text. This is especially useful when restricting negative emotions like anger. Rather than just filtering on a large raw score of anger, percent scores allow the moderator to restrict the amount of anger displayed relative to other emotions. For completeness, the graph of percent scores is shown below in figure 3.3.

In this case, the positive percent score is 100%, but in more real-world scenarios,

Figure 3.3: Percent scores applied to connotations from previous example.



the percent scores of both the connotations and emotions are more evenly distributed.

3.2.3 Emotions vs. Connotations

The motivation behind making a distinction between "emotions" and "connotations" in the lexicon is to separate them in calculating the percent scores. Positive and negative are connotations rather than emotions, so including the raw negative score would bring down the percent score of anger, for instance. From the lexicon, positive and negative act more as sentiments that are provided alongside the emotional data. While they do not interfere when calculating raw scores, the distinction became necessary when calculating percent scores. Because of this, the percent scores of positive and negative will always sum to 100%, as will the percent scores of the emotions. This is especially important when implementing tolerance thresholds, which will be covered in section 3.3.2.

3.3 Moderation

Section 1.3 covered the current two stages of moderation, both of which are implemented in this project to better simulate a real moderation workflow. The first stage moderation is automatically handled using the npm package "bad-words". The second stage moderation is implemented by applying emotional analysis on input text and then comparing the raw and percent scores against a user-created moderation configuration file that defines tolerances for each emotion/connotation in order to determine violations. Posts that are detected as profane from the first stage or contain violations in the second stage are automatically rejected.

3.3.1 First stage

By, using the npm package "bad-words", input text is easily scanned for bad language before emotional analysis is performed. This package tokenizes input into words and does a case-insensitive match against two word lists. The first word list contains profanity including variants that have numbers or symbols replacing letters (e.g. '3' instead of 'E', or '$' instead of 'S'). The second list contains additional profanity scraped from the profanity filter used in Google's "what do you love" project. Put together, these lists cover a wide variety of profanity and slurs to greatly reduce the amount of toxic posts that even reach the second stage of moderation.

3.3.2 Second stage

In the second stage of moderation, emotional analysis is performed as described in section 3.2. Once both raw scores and percent scores have been calculated, the results are then checked against a moderation configuration file. For this project, the configuration file was implemented as top-level JSON file in the project directory named "moderation-config.json". The moderation configuration supports enforcing minimum and maximum values for percent scores or raw scores for all emotions and connotations. By default, no restrictions are put in-place unless specified in the file. An example configuration file is shown in listing 1. Note that enforcing a maximum percent score of "negative" is equivalent to enforcing a minimum percent score of "positive". To turn off all emotional analysis thresholds, simply set the contents of the configuration file to an empty JSON object. When a post is submitted, a moderation report is constructed that includes all first and second stage violations. This report is displayed in the error message to the user in order to explain why the post was rejected.

3.4 User Interface

The user interface is implemented using EJS templating to display a dynamic feed of posts whenever a user navigates to the home page. The site is styled using Bootstrap CSS classes. The home page displays posts in reverse chronological order (newest first) below a submission box that asks for a title and body in order to submit a post. The distinction of the title field vs. the body field is made purely to improve readability of posts. From the moderation perspective, both the title and body are included in the calculations. The title and body are joined into one input text before calculating the raw scores and percent scores. The site will reject submissions in

```
{
    "anger": {
        "rawMax": 0.28,
        "percentMax": 35
    },
    "disgust": {
        "rawMax": 0.28,
        "percentMax": 35
    },
    "fear": {
        "rawMax": 0.36,
        "percentMax": 45
    },
    "joy": {
        "rawMin": 0.05,
        "percentMin": 10
    },
    "positive": {
        "rawMin": 0.15,
        "percentMin": 40
    },
    "negative": {
        "rawMax": 0.4,
        "percentMax": 60
    }
}
```

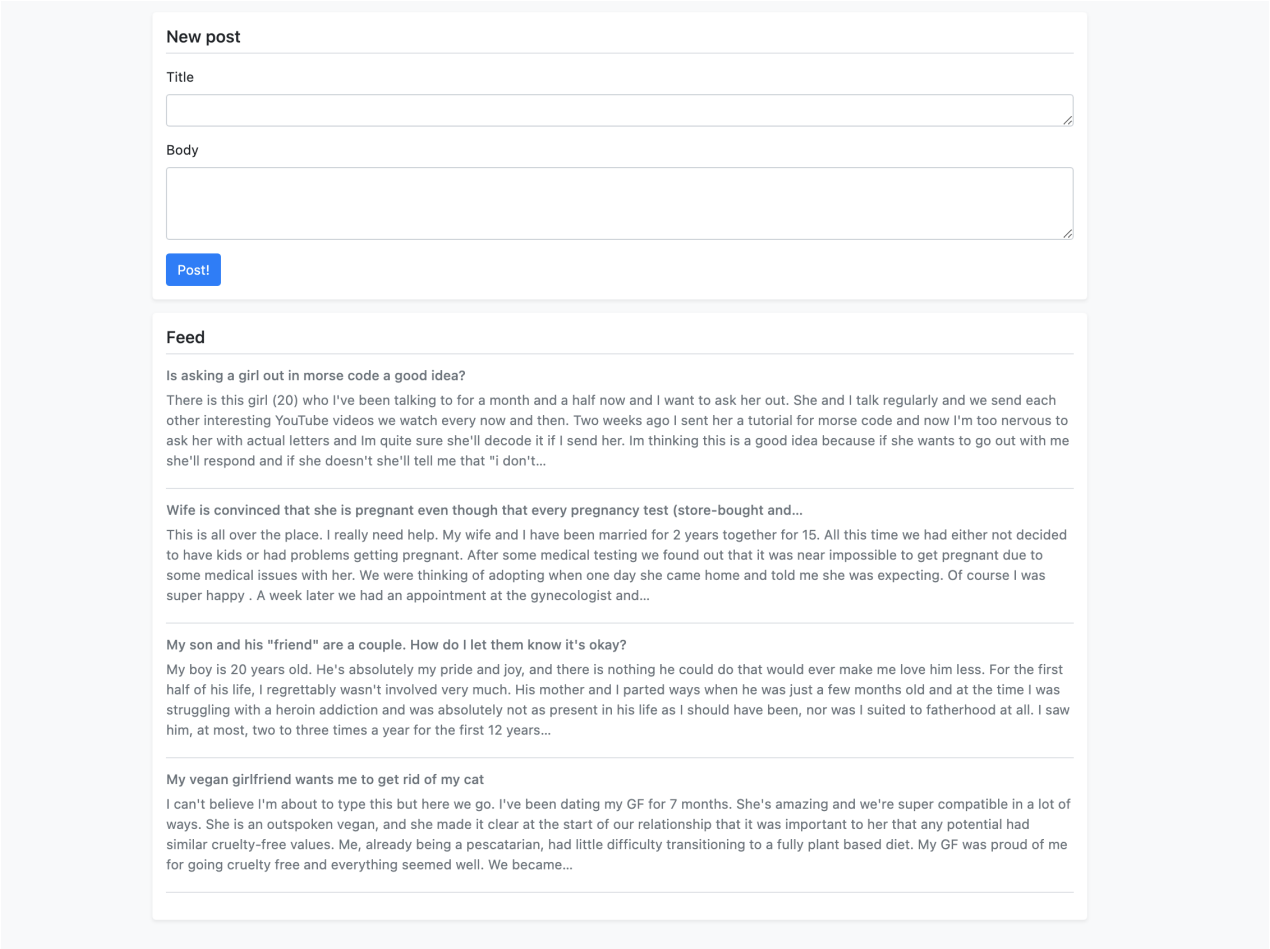
Listing 1: Example moderation configuration

which either the title or the body are blank. Additionally, there is a maximum length for both of these fields to prevent excessively large posts. The home page is shown in figure 3.4. The site contains sample submissions comprised of real-world posts containing a mix of positive and negative tones.

Clicking on a particular post in the feed leads to a new page with post details. This information is also dynamically rendered with EJS. All posts are indexed by a universally unique identifier (UUID). Post details can be viewed on the site by clicking on the post or navigating to "/post/<id>" where "id" is a UUID associated with a particular post. On the post details page, the full title and body are displayed as well as the result of the emotional analysis that was performed on the post. There is also a link back to the main feed of posts and a button to delete the post. The emotional analysis results are displayed in two tables, one displaying raw and percent scores for the emotions, and the other displaying raw and percent scores for the connotations. The full post details page can be seen in figure 3.5 with an example post.

As mentioned previously, posts which contain violations in either the first or second stage of moderation are not allowed. The specific violations are displayed to the user

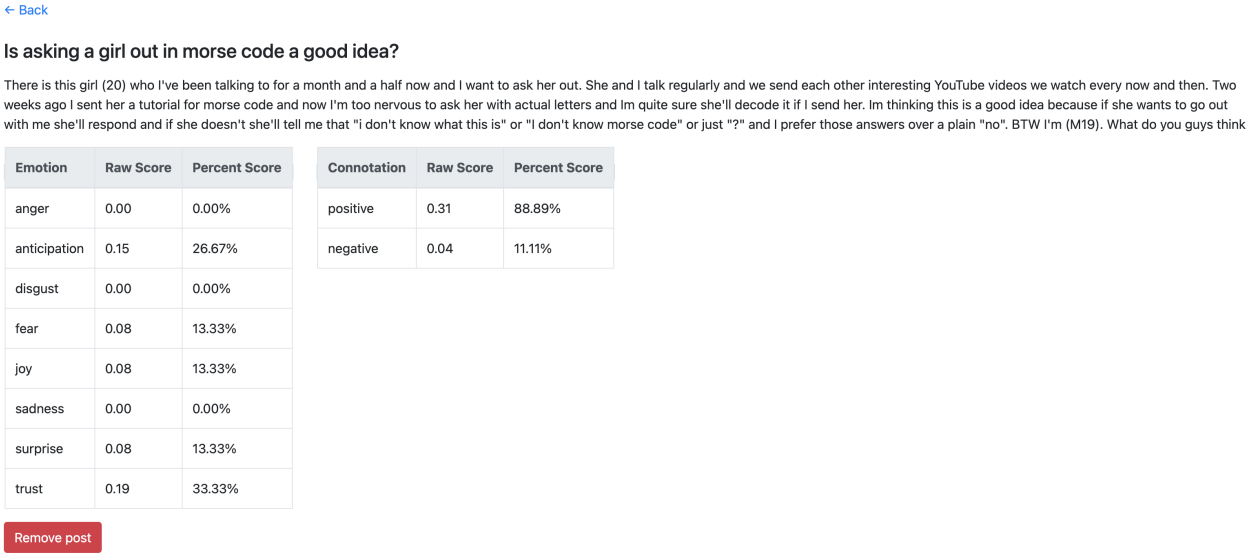
Figure 3.4: The home page of the site, displaying a submission form and feed of recent posts



upon rejection of a post. The violations are expressed as JSON. A sample violation response that contained violations in both the first and second stage is shown in listing 2.

Note that in the violation response, the expected range is provided for each emotion and connotation that contains a violation. The upper and lower bounds are defined from the moderation configuration mentioned in section 3.3.2.

Figure 3.5: The post details page, displaying the full text post along with its emotional analysis scores.



← Back

Is asking a girl out in morse code a good idea?

There is this girl (20) who I've been talking to for a month and a half now and I want to ask her out. She and I talk regularly and we send each other interesting YouTube videos we watch every now and then. Two weeks ago I sent her a tutorial for morse code and now I'm too nervous to ask her with actual letters and Im quite sure she'll decode it if I send her. Im thinking this is a good idea because if she wants to go out with me she'll respond and if she doesn't she'll tell me that "i don't know what this is" or "I don't know morse code" or just "?" and I prefer those answers over a plain "no". BTW I'm (M19). What do you guys think

Emotion	Raw Score	Percent Score
anger	0.00	0.00%
anticipation	0.15	26.67%
disgust	0.00	0.00%
fear	0.08	13.33%
joy	0.08	13.33%
sadness	0.00	0.00%
surprise	0.08	13.33%
trust	0.19	33.33%

Connotation	Raw Score	Percent Score
positive	0.31	88.89%
negative	0.04	11.11%

Remove post

```
{
   "error": "This is not an acceptable post",
   "violations": [
      "Post contains profanity.",
      {
         "emotion": "anger",
         "expectedRange": "[0, 0.28]",
         "actual": "1.00"
      },
      {
         "emotion": "anger",
         "expectedRange": "[0%, 35%]",
         "actual": "53.85%"
      },
      {
         "emotion": "disgust",
         "expectedRange": "[0, 0.28]",
         "actual": "0.57"
      }
   ]
}
```

Listing 2: Example violation response upon post submission

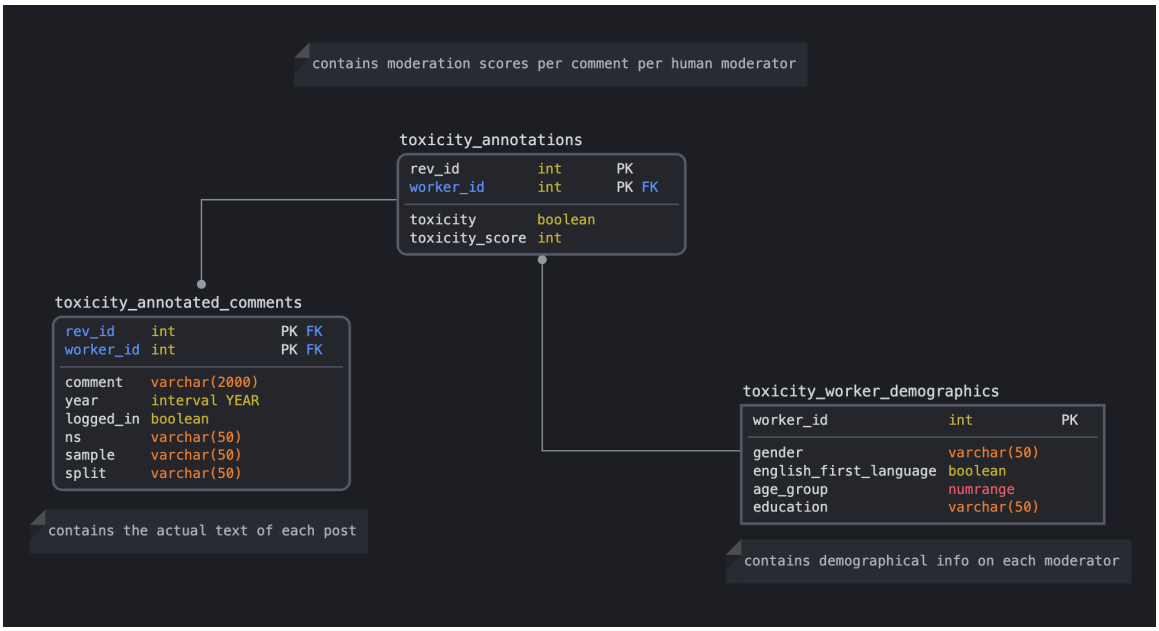
Chapter 4

Methodology/Evaluation

This project was tested using a dataset of real-world posts submitted to Wikipedia, each of which has been reviewed by a group of human moderators and scored for its toxicity. [2] This dataset contained 159,687 articles of which 39409 (24.68%) were rated negatively by the moderators indicating that they should be filtered and 120278 (75.32%) were rated positively. The dataset was originally provided in the form of three tab-delimited files. The first file in the dataset contained the text of each comment along with metadata with additional details about the post like the year it was posted. This file also contains id‘s for every post called the “rev_id”. The second file contains the results of the human moderators manual analysis of every post. Every entry starts by referencing the post using the same “rev_id” and then contains one human moderator‘s decision. This decision is denoted firstly by a boolean field “toxicity” and secondly by a scalar field “toxicity_score”. As each post was reviewed by multiple human moderators, the first file has a “one-to-many” relationship with the second file. The toxicity score field indicates the varying degrees of toxicity. A 0 score indicates a neutral post, whereas increasingly negative numbers indicate increasing degrees of toxicity. Increasing positive numbers indicate decreasing degrees of toxicity. For this project, only the toxicity score was used. The third file contained demographic information on the human moderators, which was ignored for this project. Figure 4.1 provides a visual of the dataset’s initial structure.

The records in the dataset were then joined using a script to get the majority decision for determining the toxicity of each post and joining this with the text of the posts into a single JSON file. The “toxicity_score” field for each human moderator‘s review were totaled to achieve a net toxicity score per post. Once combined, a value of 0 or greater is interpreted as not toxic, while a negative score is interpreted as toxic. The assumption for the rest of the testing process assumes that a negative toxicity score indicates that a post should ideally result in violations when emotion

Figure 4.1: Initial structure of the dataset



analysis is applied in the second stage moderation. Listing 3 shows the structure of the dataset after the majority votes have been calculated and merged into one JSON file along with the body of each post. Note that the "id" refers to the "rev_id" from the initial structure. While it is not necessary in the new structure, it was included to easily reference specific posts before and after the reworking of the dataset structure for debugging purposes.

Once the dataset had been reworked, a test script was created to iterate through each post in the dataset and check for profanity before performing emotional analysis and violation checks against the supplied moderation configuration. If a particular post was considered toxic by the consensus of the human moderators as well as being in violation of either the first or second stage of moderation, then the post was marked as a true positive. Likewise, if the post was evaluated by the human moderators as not being toxic, then a true negative result required no violations in either stage of moderation. If there was a disagreement between the human moderators and the moderation stages of this project, then a failure was noted for that post. This includes both false positives and false negatives. After performing this operation on every post in the dataset (approximately 150,000 posts), then the accuracy was calculated as the percentage of true positives and true negatives out of the total number of posts that were evaluated. From the dataset, 75.32% of posts were not labeled as toxic by the moderators. This establishes a simple baseline to compare against. Section 4.1 discusses the results and the iteration process of modifying the moderation configuration to achieve a better accuracy.

```
[
  ...
  {
    "id": "146800286",
    "toxicityScore": 2,
    "comment": "<non-toxic post here>"
  },
  {
    "id": "146801042",
    "toxicityScore": -3,
    "comment": "<toxic post here>"
  },
  ...
]
```

Listing 3: Evaluation dataset structure after modification

4.1 Results

With a test script in place and a transformed dataset for evaluation, the script was run against a simple moderation configuration that restricted the maximum raw and percent scores for anger, disgust, fear, and negative. The configuration also included mandatory minimums for joy and positive. The first result was 65% accuracy. Following the first result, the test script was modified to include additional logging to provide insights into which violations were occurring in the second stage when the human moderators decided the post was not toxic. This case where second stage is "too strict" is considered a false positive, or a type I error. In the case of being "too lenient" and incurring no violations on a post that the human moderators identified as toxic, the raw scores were added to a running average to identify the average raw scores for posts that were false negatives. These are considered type II errors, where posts were wrongly accepted despite actually being toxic according to the human moderators. Listing 4 shows an example of the logging after the evaluation script is run. Using these metrics, the moderation configuration was manually modified in incremental changes to improve the accuracy. It was found that enforcing minimums in positive emotions/connotations resulted in type I errors (false positives) against neutral posts. After removing these restrictions and raising the maximums for negative emotions/connotations, the amount of type I errors reduced significantly, raising the accuracy to around 72%. Following this, the maximums for negative emotions/connotations were continually changed until the restriction on the "negative" connotation was removed altogether. This removal also reduced type I errors and brought the accuracy to 81.45%. With those changes in place, the amount of type I

```
{
  "tooLenient": {
    "occurences": 22532,
    "averageEmotionScores": {
      "anger": 0.039692790311154844,
      "disgust": 0.031230551938189993,
      "fear": 0.043604763573512235,
      "negative": 0.13900089446966934
    }
  },
  "tooStrict": {
    "anger": 18628,
    "disgust": 13342,
    "fear": 9483,
    "negative": 14560
  }
}
```


Listing 4: Example logging of the evaluation script

and type II errors were roughly equivalent. Listing 5 shows the moderation configuration that achieved an 81.45% accuracy. Listing 6 shows the logging produced by the evaluation script for this configuration. The number of type I and type II errors are approximately the same. The full confusion matrix is shown in table 4.1, and the equation used to calculate accuracy is shown in equation 4.1.


```
{
    "anger": {
        "rawMax": 0.28,
        "percentMax": 35
    },
    "disgust": {
        "rawMax": 0.28,
        "percentMax": 35
    },
    "fear": {
        "rawMax": 0.36,
        "percentMax": 45
    }
}
```


Listing 5: Moderation configuration that achieved an accuracy of 81.45%.

$$\text{accuracy} = 100 \times \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{true negative} + \text{false positive} + \text{false negative}} \tag{4.1}$$

```
{
  "tooLenient": {
    "occurences": 14142,
    "averageEmotionScores": {
      "anger": 0.039692790311154844,
      "disgust": 0.031230551938189993,
      "fear": 0.043604763573512235,
      "negative": 0.13900089446966934
    }
  },
  "tooStrict": {
    "anger": 7498,
    "disgust": 2861,
    "fear": 7411
  }
}
```


Listing 6: Evaluation script logging for the configuration shown in listing 5.

Table 4.1: Confusion Matrix for the configuration that achieved an 81.45% accuracy out of the 159,687 total posts.

		Actual	
		Positive	Negative
Predicted	Positive	15,315	15,487
	Negative	14,142	114,743

Total: 159,687

Chapter 5

Conclusion

The work outlined in this paper is for the purpose of assisting moderators online in identifying and moderating toxic posts. The moderation tools developed as part of the work on this paper reduce the amount of posts that require human review by preemptively moderating posts using an emotional analysis algorithm in conjunction with a moderator-supplied configuration that defines tolerances of each emotion. Several different configurations were tested. The highest accuracy among them was 81.45%, compared to a baseline of 75.32%, which is obtained by classifying all posts in the dataset as non-toxic. Most likely, a machine learning approach is needed to determine the optimal configuration, however the evaluation work as part of this project stops here.

5.1 Improvements/Future Work

With an 81.45% accuracy compared to the baseline of 75.32%, the second stage moderation that was the focus of this project could definitely be improved in a number of ways. As mentioned in section 3.2.1, the emotional analysis algorithm treats each word in isolation when determining both the raw scores and percent scores for a given input text. As a result, the algorithm fails to account for modifiers like “not” or “very”. The overall accuracy could be improved by adjusting the emotional analysis algorithm to contain a look-ahead window that amplifies or negates the scoring of words preceded by a modifier. A more flexible approach would be to employ a neural network trained by part of the testing data. As shown in figure 4.1, the comments are classified under a particular “split” value. These values are “train”, “test”, and “dev”. By training a neural network with only the “train” data, over-fitting could be avoided when testing against the “test” split data. A neural network could learn to recognize patterns in complex speech that tie words or phrases together to achieve

overall emotions that would be undetectable when treating each word in isolation.

Beyond the emotional analysis algorithm, the project as a whole could be improved by adding the option to allow posts that incur violations in the second stage moderation but flag these posts for manual review. This would help reduce the type I errors (false positives) while still reducing the workload of human moderators from reviewing every submission to only reviewing flagged submissions. Another improvement would be making the underlying process of performing emotional analysis and checking for violations against a user-supplied configuration file more accessible via a distributable package rather than a standalone proof of concept that is tied into an existing project. If the core of the emotional analysis and moderation were decoupled from the larger project, it could be made available on package registries like NPM or Maven Central. This would allow other administrators or developers to easily integrate the work into their own projects.

5.2 Contribution

To achieve the goals of this project mentioned in section 1.4, I developed the website shown in section 3.4. As mentioned in the acknowledgements (section 5.3), Andrew Mackey provided starter code written in Java for the emotional analysis algorithm. I worked on porting this algorithm to Javascript, as the back-end of the website was written in Javascript. After setting up the moderation tool to detect violations, I spent the remainder of my time on this project trying to improve the accuracy of identifying toxic posts by altering the configuration and extracting metrics to identify the number of type I and II errors.

5.3 Acknowledgements

The focus of this project takes inspiration from the work of PhD candidate Andrew Mackey. His guidance in the work of this project involved supplying the datasets used in the emotional analysis algorithm and in testing the effectiveness of the project at classifying toxic input text. He also provided starter code for the emotional analysis algorithm. Mackey's ongoing work involves the application of emotional analysis to detect "fake news". Additionally, this project would not have been possible without the support of Dr. Susan Gauch, who oversaw the work and provided guidance and direction throughout the process.

Bibliography

[1] Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. "Assessing State-of-the-Art Sentiment Models on State-of-the-Art Sentiment Datasets". In: *CoRR* abs/1709.04219 (2017). arXiv: 1709.04219. URL: http://arxiv.org/abs/1709.04219.

[2] *ConversationAIDataset*. Kaggle. 2018. URL: https://www.kaggle.com/eoveson/conversationaidataset.

[3] Charles Darwin. *The expression of the emotions in man and animals*. Oxford University Press, 1872.

[4] Paul Ekman. *Facial Expression and Emotion*. 1992. URL: https://sanlab.psych.ucla.edu/wp-content/uploads/sites/31/2016/03/Ekman-American_Psychologist_1993.pdf.

[5] Nida Hakak et al. "Emotion analysis: A survey". In: July 2017, pp. 397–402. DOI: 10.1109/COMPTELIX.2017.8004002.

[6] Shagun Jhaver et al. "Human-Machine Collaboration for Content Regulation: The Case of Reddit Automoderator". In: 26.5 (2019). ISSN: 1073-0516. DOI: 10.1145/3338243. URL: https://doi.org/10.1145/3338243.

[7] Andrew Lee Mackey. "Detecting Fake News through Emotion Analysis". (In Preparation).

[8] Alexis C. Madrigal. *Inside Facebook's Fast-Growing Content-Moderation Effort*. The Atlantic. Feb. 2018. URL: https://www.theatlantic.com/technology/archive/2018/02/what-facebook-told-insiders-about-how-it-moderates-posts/552632/.

[9] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey". In: *Ain Shams Engineering Journal* 5.4 (2014), pp. 1093–1113. ISSN: 2090-4479. DOI: https://doi.org/10.1016/j.asej.2014.04.011. URL: https://www.sciencedirect.com/science/article/pii/S2090447914000550.

[10] Saif M Mohammad and Peter D Turney. *NRC Word-Emotion Association Lexicon*. 2013.

[11] Carlo Strapparava and Rada Mihalcea. "Learning to Identify Emotions in Text". In: SAC '08. New York, NY, USA: Association for Computing Machinery, 2008. ISBN: 9781595937537. DOI: 10.1145/1363686.1364052. URL: https://doi.org/10.1145/1363686.1364052.