Data Profiling and Quality

1. Profiling – Data Sources

Two types of data are used in this project which are referred to as *base data* and *augmented data*.

1.1 Base data

The base data is obtained from the ongoing kaggle competition titled "*Natural Language Processing with Disaster Tweets*" and is currently available at:

https://www.kaggle.com/competitions/nlp-getting-started/data

The base data consists of two files: **train.csv** and **test.csv**. The data in these files will be referred to as *training data* and *testing data* respectively and have the following characteristics:

train.csv - This file has 8562 lines of raw text. Each sample (row) has the following 5 fields:

- **id -** integer, unique identifier for each row which should always have a value
- **keyword -** string, a particular keyword from the tweet which may be blank
- **location -** string, the location the tweet was sent from which may be blank
- **text -** string, the text of the tweet
- **target -** integer, 1 or 0 representing a binary label to be classified and denotes whether a tweet is about a real disaster that needs to be responded to (1) or not (0)

test.csv - This file has 3700 lines of raw text. Each test sample (row) has the same first 4 fields (columns) as the train.csv file: **id**, **keyword**, **location** and **text**. There is no **target** column as these values are expected to be predicted by participants of the competition and scored by kaggle.

1.2 Augmented Data

The augmented data will consist of tweets generated from a large language model (LLM) prompted to create something similar to each tweet in the **text** column of the train.csv file. The prompt used to generate tweets will differ depending on which class label the generated tweet was based on.

The two tables provided in Appendix A show samples of tweets generated from the ChatGPT 3.5 (ChatGPT) website. The first table of Appendix A shows prototype results that were generated from a prompt designed to create disaster-related tweets (target = 1). The results shown in the second table were generated from a prompt designed to create NOT-disaster-related tweets (target = 0). Data for both tables were manually collected on 2024-02-22.

2. Profiling – Variables

2.1 Text and target data

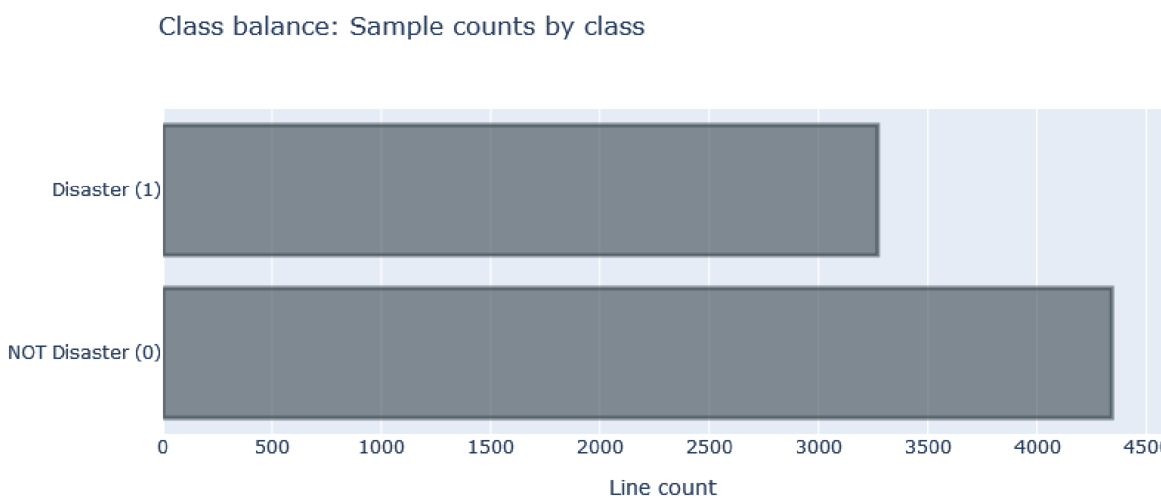
Of the 5 fields described in the section 1.1, only the **text** and **target** fields are being used in the analysis. The **id** field is simply an identifier. No **id** values are missing or duplicated. Since the **id** field is not used to identify duplicates, it will only be used to identify samples.

The **keyword** and **location** fields could be used as model predictors, but because the objective of the analysis is to evaluate the impact of LLM data augmentation, there was no compelling reason to keep them.

There are no missing **text** or **target** values (see Appendix B). However, there are duplicate tweets as described in Appendix C. Because they represent a small portion of the overall training data, duplicates will be removed before augmenting the data prior to model training.

2.2 Class balance

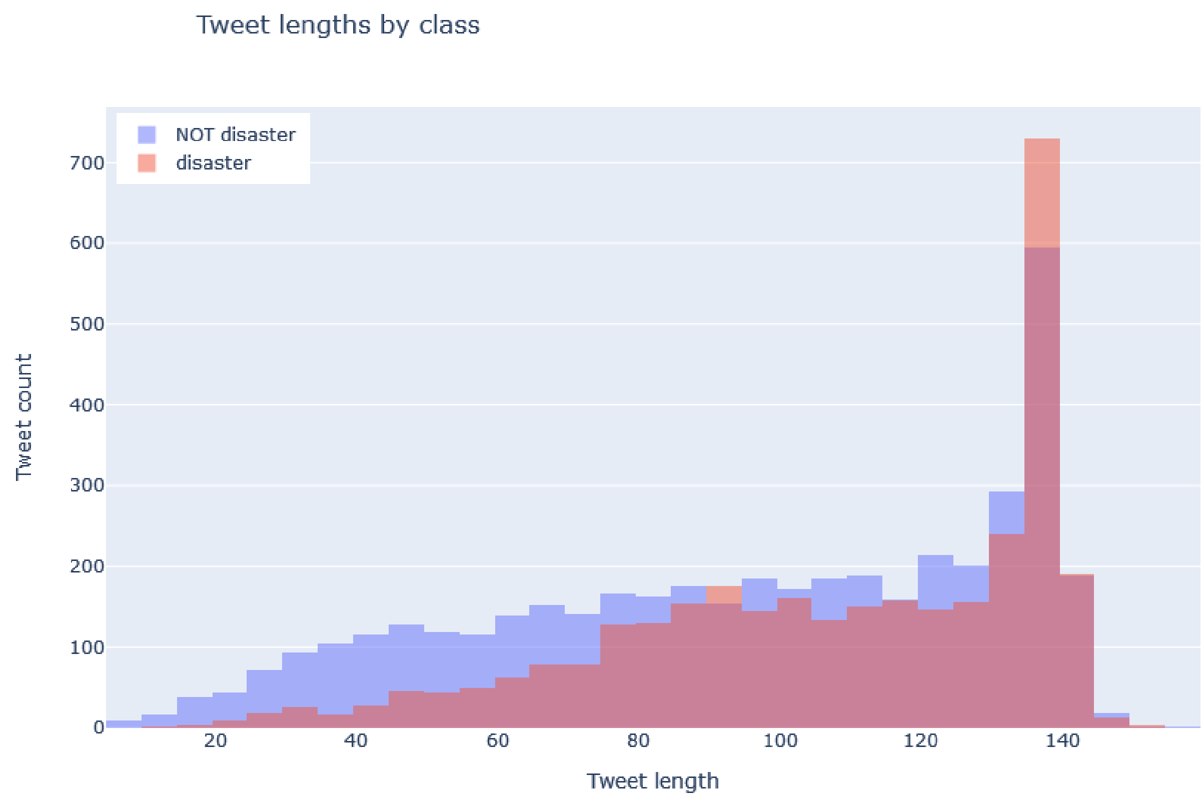
Severe class imbalance can make it difficult for a model to learn the minority class. Fortunately, the classes are roughly similar in sample counts as shown in the figure below.



2.3 Tweet length by class

Disaster-related tweets (orange) appear to have more probability mass towards longer tweet lengths and not-disaster tweets (blue) have more mass towards the shorter tweet lengths. This is shown in the next figure where the red mass indicates overlap of the two distributions. This implies that disaster-related tweets will tend to be a little longer on average than non-disaster tweets.

Predictor variables such as tweet length and URL count could be used as features in future modeling projects where the primary goal is accuracy. However, they are not considered in this project because they don’t contribute to the goals of this project.



2.4 Interesting observations so far – inconsistent responses and sentiment flipping

Two issues emerged while looking at the tweets generated from prompts to ChatGPT. The first issue was that the results were not repeatable even when the same prompt was provided. For example, when the following prompt for a NOT disaster tweet was given to ChatGPT on 2024-02-22:

```
write me a tweet similar to this one, under 141 characters, but
refers to a different activity and location:

Est. September 2012 - Bristol,We always try to bring the heavy.
#metal #RT http://t.co/YAo1e0xngw
```

The following response was received:

Founded in June 2015 - Seattle, We're here to rock hard and melt faces. 🤘🎸 #rocknroll #SeattleMusic

When the same prompt was given on 2024-02-23, ChatGPT responded with:

Established September 2015 - New Orleans, We consistently deliver the spicy. #jazz #NOLA

The second issue observed was that sometimes that sentiment polarity of the response "flipped" to the other class or became less obvious. For example, when the following prompt for a disaster tweet was given to ChatGPT on 2024-02-22:

```
write me a tweet similar to this one, under 141 characters, does not
use contractions, but refers to a different disaster and location:

Our Deeds are the Reason of this #earthquake May ALLAH Forgive us all
```

The following response was received:

Our actions shape our world; let's unite in prayer as floods devastate Indonesia. May we find forgiveness and strength. 🙏 #IndonesiaFloods

When the same prompt was given on 2024-02-23, ChatGPT responded with:

The actions we take shape our world. Let's seek forgiveness for our impact. #reflection #peace

This more current response does not imply that a natural disaster had occurred which is considered an error because the associated label with this tweet is target = 1 or disaster-related.

2.5 Prompt design and ChatGPT 4

The only way to reliably quantify the extent of errors introduced when sentiment flips to the unintended class is to manually curate responses from ChatGPT. Since it is impractical to manually evaluate approximately 8000 augmented tweets, an estimate of the proportion of mislabeled tweets from the augmented training samples will be made on samples of this data.

The results of taking a small sample of tweets using each prompt described in Appendix A (26 + 25 = 51 total) suggests that sentiment polarity will flip on average 4 % (1 in 26) of the time when requesting a tweet similar to one that is disaster-related (target = 1) and 0 % (0 in 25) when requesting a tweet similar to one that is not disaster-related (target = 0) as described in Appendix D.

The prompts described in Appendix A were the result of a number of trial-and-error iterations. After several of these ad hoc iterations, the responses coming back from ChatGPT subjectively appeared to be more closely aligned with the intended sentiment. Because these error are expected to improve as a result of improving the prompt design, further iterations on these prompts may be made throughout the remainder of the project as time permits.

ChatGPT 3.5 is initially being explored in this project because it is much cheaper to use than ChatGPT 4 [2]. Version 4 is much more powerful than 3.5 and would likely generate augmented tweets with a lower error rate. If time permits and it doesn't seem to be too costly, using version 4 may be explored.

2.6 Additional challenge – tokenization

Tokenization is the breaking down of unstructured text into pieces that can be vectorized. Tokens are typically words or word fragments that can be counted or mapped to a word embedding such as GloVe [1]. Doing tokenization properly is a challenge common to all NLP tasks regardless of data quality. This challenge is more acute when working with tweets because users adopt abbreviated communication patterns in response to the character limitations of the platform.

2.7 Metrics

Since the goal of the project is to evaluate whether LLM generated data can improve another model's performance, the accuracy and the area under the receiver operating curve (ROC AUC) of the trained model with and without LLM augmented data are the metrics selected to evaluate whether the research hypothesis (see project proposal) is true or not.

2.8 Profiling summary

- Two types of data are utilized in this project: base data and augmented data
- The source of the base data is https://www.kaggle.com/competitions/nlp-getting-started/data
- The source of the augmented data is ChatGPT 3.5, but version 4 may be explored.
- Base training data has 5 fields of which only 2 will be used for the project: **text** and **target**
  - **text** - contains the text of the tweet
  - **target** - the response being modeled, 1 = disaster related, 0 = NOT disaster related
- There are no missing text or target values
- target = 0 and target = 1 samples make up 57% and 43% of the total data respectively, so class imbalance issues are not expected to be a problem.
- Disaster-related tweets tend to be a little longer than NOT disaster tweets on average.
- Two interesting observations have come up so far: ChatGPT response inconsistency and sentiment polarity flipping
- ChatGPT response inconsistency by itself should not pose any problems with this analysis.
- Sentiment polarity flipping is a problem that needs to be acknowledged.
  - Eliminating these types of errors requires manual review by a human which is impractical to perform on the entire augmented data set given the number of samples (roughly 8000), resources (just me) and schedule.
  - Initial estimates based on a small sample of augmented tweets indicates that polarity flipping occurs about 4% of all target = 1 samples while no target = 0 samples had flipped.
  - Although these errors will degrade model performance, the magnitude of this degradation relative to the potential enhancement provided by augmentation is unknown at this time.
  - Improved prompt design is likely to reduce sentiment flipping errors as would using ChatGPT 4 instead of 3.5 to generate the augmented data.
- The base and augmented data appear to be sufficient to adequately test the research hypothesis.

3. Quality

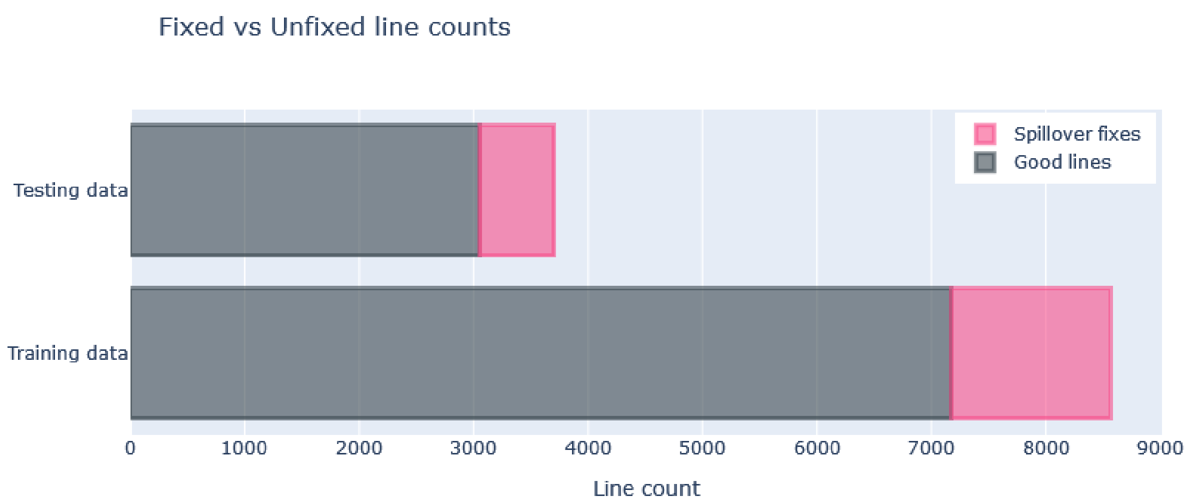
There are two main data quality issues with the base data: data formatting and duplicate records. The data formatting problem arises from the observation that each sample should be on one line but is not. The duplicate records problem arises from the observation that tweets in the text field are not unique.

3.1 Data Formatting

The training and testing data both contain formatting problems which needed to be addressed before tokenization can be done. This problem can be observed in both the training and testing data as “spillover lines”. A couple of examples of these problems shown below:

```
id,keyword,location,text,target
61,ablaze,,"on the outside you're ablaze and alive
but you're dead inside",0
74,ablaze,India,"Man wife get six years jail for setting ablaze niece
http://t.co/eV1ahOUCZA",1
86,ablaze,Inang Pamantasan,"Progressive greetings!
```

These type of problems are fixed with the `fix_spillover_lines` function of the `projtools.py` module of the project repository [3]. The figure below shows the number of lines this function repaired on the base training and test data.



3.2 Duplicate data

Duplicate data can be defined in a number of ways. For example, when a tweet is retweeted by another user, the tweet text is typically the same as the original tweet. In this case, the tweet is technically not a duplicate, even though the content is the same as the original.

For the purposes of this project, tweets with the same **text** and **target** content are considered duplicates since we are only using the text field to generate model features. These rows (110) were removed from the training data.

There are a small number of training samples (36) that have the same text content, but with a different target label. Because these samples are obviously problematic as they present conflicting information to the model during training, they were also removed from the training data.

3.3 Quality summary

- Most base data samples (84% of training, 82.5% of testing) reside on a single row.
    - Data that spilled over into additional lines were fixed with a Python function (**fix_spillover_lines** provided in the **projtools.py** module of the project repo)
- A small number of duplicate data samples were found (110 + 36 = 146). These samples were removed.
    - Because these were a small portion of the total training data, the impact of removing these samples is assumed to be negligible.

Appendix A – Prototypes of augmented data

The prompt designed to create the augmented data for disaster class samples is as follows:

`write me a tweet similar to this one, under 141 characters, but refers to a different` <u>`disaster`</u> `and location:  <tweet from train.csv>`

The table below shows responses from the ChatGPT 3.5 website (https://chat.openai.com) to the above prompt on four tweets from the disaster class (target = 1) of **train.csv** inserted into the "**`<tweet from train.csv>`**" text.

PROMPT	CHAT3.5 RESPONSE
write me a tweet similar to this one, under 141 characters, but refers to a different disaster and location: *Our Deeds are the Reason of this #earthquake May ALLAH Forgive us all*	*"Our actions shape our world; let's unite in prayer as floods devastate Indonesia. May we find forgiveness and strength. 🙏 #IndonesiaFloods"*
write me a tweet similar to this one, under 141 characters, but refers to a different disaster and location: *Forest fire near La Ronge Sask. Canada*	"*Heart goes out to those impacted by the flood in Houston. Sending support and strength to all affected. Stay safe. 🙏💧 #HoustonFlood*"
write me a tweet similar to this one, under 141 characters, but refers to a different disaster and location: *All residents asked to 'shelter in place' are being notified by officers. No other evacuation or shelter in place orders are expected*	*"All residents urged to 'shelter in place' in Miami Beach due to ongoing hurricane warnings. Stay indoors and stay safe, everyone. 🌀 #MiamiBeach #HurricanePreparedness"*
write me a tweet similar to this one, under 141 characters, but refers to a different disaster and location: *13,000 people receive #wildfires evacuation orders in California*	*"Urgent: 15,000 residents under evacuation orders due to flooding in New Orleans. Stay safe and heed official warnings. 🌊 #NewOrleans #FloodSafety"*

The prompt designed to create the augmented data for NOT disaster samples is as follows:

`write me a tweet similar to this one, under 141 characters, but refers to a different` <u>`activity`</u> `and location:  <tweet from train.csv>`

The following table shows responses from the ChatGPT 3.5 website (https://chat.openai.com) to the above prompt on four tweets from the NOT disaster class (target = 0) of **train.csv** inserted into the "`<tweet from train.csv>`" text.

Appendix A (cont.)

PROMPT	CHAT3.5 RESPONSE
write me a tweet similar to this one, under 141 characters, but refers to a different activity and location: *Est. September 2012 - Bristol,We always try to bring the heavy. #metal #RT http://t.co/YAo1e0xngw*	*"Founded in June 2015 - Seattle, We're here to rock hard and melt faces. 🤘🎸 #rocknroll #SeattleMusic"*
write me a tweet similar to this one, under 141 characters, but refers to a different activity and location: *Crying out for more! Set me ablaze*	*"Yearning for adventure! Ignite my soul."*
write me a tweet similar to this one, under 141 characters, but refers to a different activity and location: *On plus side LOOK AT THE SKY LAST NIGHT IT WAS ABLAZE http://t.co/qqsmshaJ3N*	*"Look up! The city skyline was aglow last night, painting a mesmerizing scene. ✨ #CityLights #UrbanBeauty"*
write me a tweet similar to this one, under 141 characters, but refers to a different activity and location: *@PhDSquares #mufc they've built so much hype around new acquisitions but I doubt they will set the EPL ablaze this season.*	*"Excitement brewing over the new lineup, but will they set the stage on fire this season? 🔥⚽ #PremierLeague #FootballFever"*

Appendix B – Missing text and target values

All spillover rows are fixed and rewritten in cells 4 of the following Python jupyter notebook:

https://github.com/MichaelSzczepaniak/llmamd/blob/main/preproc_disaster.ipynb

This cell uses the `fix_spillover_lines` function listed in the projtool.py module located at:

https://github.com/MichaelSzczepaniak/llmamd/blob/main/projtools.py

Appendix C – Counting duplicate tweets

Duplicate tweets are processed in cells 8 through 13 of the same Python jupyter notebook listed in Appendix B:

https://github.com/MichaelSzczepaniak/llmamd/blob/main/preproc_disaster.ipynb

Appendix D – Estimating augmented data sentiment flip

Raw data for these results can be found at:

https://github.com/MichaelSzczepaniak/llmamd/blob/main/data/2022.02.24.chatgpt.csv

References

[1] https://nlp.stanford.edu/projects/glove/
[2] https://openai.com/pricing
[3] https://github.com/MichaelSzczepaniak/llmamd/blob/dev/projtools.py
[4] https://stats.stackexchange.com/questions/156796/how-to-build-a-confidence-interval-with-only-binary-test-results#156807