

# Classification of Tweets Related to the 2022 Russian Invasion of Ukraine

Andy Gnias

May 9, 2022

# 1 Abstract

Social media applications such as Twitter are used extensively by reporters for producing micro-reports, and by users of all types to express their opinions on current events. The 2022 Russian Invasion of Ukraine saw a massive influx of Tweets related to the conflict. This paper presents several methods used in an attempt to classify Tweets related to the conflict into discrete categories. Data was obtained using Twint, a Twitter data mining tool, to pull Tweets relevant to the conflict. Sentiment Analysis, k-means clustering, and a Neural Network were used to label these Tweets, and results for each method were compared.

# 2 Introduction

On February 24, 2022, Russian forces invaded Ukraine in what was a major escalation to a conflict that had been ongoing since 2014[3]. It has since been the primary focus of many news outlets, and has led to many to express their opinions on the conflict via Twitter. The goal of the research conducted in this paper was to determine which methods are feasible in accurately determining whether a Tweet is expressing an opinion in favor of, or against, one side of the conflict or another. Developing accurate classification methods could allow for an overall sentiment on the conflict to be determined, or for an individual Twitter user's views to be categorized.

Specifically, these methods were used to classify a Tweet into one of seven categories: Pro/Anti-Ukraine, Pro/Anti-Russia, Neutral, expressing an opinion on some other entity, or uncategorizable. Methods that are successful in this domain can also be applied to virtually any other issue, so long as the data can be acquired in a similar fashion.

Three methods in the domain of Data Mining were utilized for classifying these Tweets: Sentiment Analysis, Clustering, and Classification via a Neural Network.

Sentiment Analysis is a Natural Language Processing technique used to determine whether text is expressing a positive, negative or neutral emotion. The objective in performing Sentiment Analysis on these Tweets is based on the idea that a positive sentiment on an aspect indicates a positive opinion of that aspect, and a negative sentiment indicates a negative opinion. The opinion expected based on the Sentiment Score of a Tweet with a particular hashtag is as follows:

- #Ukraine and #Zelensky - A positive sentiment indicates a Pro-Ukraine Tweet and a negative sentiment indicates an Anti-Ukraine Tweet
- #Russia and #Putin - A positive sentiment indicates a Pro-Russia Tweet and a negative sentiment indicates an Anti-Russia Tweet
- Tweets from any category with a sentiment score of 0 indicates a Neutral sentiment

Clustering attempts to group similar objects in a dataset together. Although not technically a classification method, the idea behind using clustering in this context is that Tweets

pertaining to a similar topic or idea will be clustered together. Therefore, a sample of Tweets in each cluster can be used to determine an overall classification for each cluster.

Neural Networks are the gold-standard for most classification problems due to their ability to handle complex and non-linear data[20]. However, unlike Sentiment Analysis and Clustering, they require a labeled dataset when used as a supervised learning model[8]. The results of Sentiment Analysis and Clustering were manually labeled and used as inputs to the Neural Network, which allowed for a classification model to be generated.

## 2.1 Limitations

The data collected was limited to English language Tweets, as it removed any complications with analyzing Tweets in multiple languages and allowed for easy interpretation of the results. Additionally, the opinions expressed were limited to Twitter users expressing their opinion with a relevant hashtag between 03/05/22 - 03/14/22. Because of the recency of the conflict, a labeled dataset of a large batch of Tweets related to the conflict was not readily available. Therefore, the Neural Network inputs were limited to Tweets manually labeled by the author. This limited the dataset in terms of size, as well as skewed the dataset towards the author’s interpretation of each Tweet.

## 3 Related Work

Sentiment Analysis has been used extensively in text mining applications. VADER, the model used in this research for determining text sentiment, was specifically developed with social media text in mind[10]. In terms of using Sentiment Analysis for labeling, Turney (2002) used a related semantic orientation algorithm to classify reviews for different entities as “recommended” or “not recommended”, with accuracy ranging from 66-84%.[22].

Clustering was used on a FiveThirtyEight (2018) dataset of 3 million Tweets produced by Russian Trolls[17]. Several researchers were able to cluster the Tweets in the dataset by category. This method showed the potential for individual clusters of Tweets to contain Tweets pertaining to the stance of one of the Desired Categories [see sec 4.1.1].

Neural Networks have been employed for all types of classification problems, text notwithstanding. Kumar et al. (2019) used a deep neural network to classify Tweets related to disasters to better coordinate humanitarian efforts[11]. Additionally, Hallac et al. (2018) used several types of Neural Networks to classify Tweets into several selected topics[9].

## 4 Methodology

### 4.1 Data

#### 4.1.1 Categorization

In order to classify Tweets, the following categories were defined:

- Pro-Ukraine (pu) - Tweets showing positive support for Ukraine
- Pro-Russia (pr) - Tweets showing positive support for Russia
- Anti-Ukraine (au) - Tweets critical of or showing negative support for Ukraine
- Anti-Russia (ar) - Tweets critical of or showing negative support for Russia
- Neutral (n) - Tweets related to the conflict but not showing support for either side
- Other-Entity (us) - Tweets pertaining to some other entity not related to the conflict (usually critical, ex. “The conflict is bad, but what about X”)
- Unknown (u) - Either not enough context, or having nothing to do with the conflict

The first five categories are the Desired Categories. These are the categories that ideally all Tweets would fall under. The last two categories are the Consequential Categories. These categories are a consequence of irrelevant data being obtained in the data acquisition methods. It was decided to include Tweets falling under these categories as part of the analysis, as they are indicative of how often hashtags related to the conflict are used to discuss unrelated issues.

These categories can also be simplified for the sake of analysis into 3 categories, either Pro-Ukraine, Pro-Russia, or neutral. In these instances, Anti-Russia Tweets are grouped with Pro-Ukraine Tweets, and Anti-Ukraine Tweets are grouped with Pro-Russia Tweets. Neutral Tweets do not need to be altered, and Other-Entity and Unknown Tweets can be discarded.

#### 4.1.2 Acquisition

Data was acquired using the Twitter Intelligence Tool (Twint)[15]. Queries were made on hashtags of the countries and leaders involved in the conflict: #Russia, #Ukraine, #Putin, #Zelensky. Tweets returned from Twint were saved to individual files for each hashtag, and these files were then filtered. Filtering involved removing Tweets not written in the English language, which was done by removing any Tweet containing a non-ASCII character. Additionally, Tweets were removed if they contained a conflicting hashtag. Therefore:

- Tweets with #Russia were removed from the #Ukraine Tweets
- Tweets with #Putin were removed from the #Zelensky Tweets
- Tweets with #Ukraine were removed from the #Russia Tweets
- Tweets with #Zelensky were removed from the #Putin Tweets

This ensured that the hashtag was more likely to be the primary subject of the Tweet. This resulted in 397,491 Tweets with #Ukraine, 156,721 Tweets with #Russia, 35,661 Tweets with #Zelensky, and 43,748 with #Putin. The timestamp of the Tweets ranged from 03/05 to 03/14. The timestamp was not taken into consideration for any of the analyses performed.

### 4.1.3 Labeled Dataset

A subset of these Tweets were manually labeled for the purpose of analyzing the results of Sentiment Analysis and Clustering, and for use in generating a Neural Network classification model. Sentiment Analysis grouped Tweets into 1 of the 5 Desired Categories [sec 4.1.1]. For each category, the top 250 Tweets with the most intense sentiment were labeled, as their lack of neutrality and nuance made them easier to classify. No specific methodology was used in selecting Neutral Tweets. Additionally, Tweets for each of the three clusters generated during k-means clustering were labeled. In each cluster, the 250 Tweets closest to the cluster centroid were labeled, as these were the Tweets that best represented the cluster. This resulted in a labeled sample of 2,000 Tweets spanning all 7 categories [sec 4.1.1].

### 4.1.4 Augmented Dataset

The labeled dataset contained significantly less examples of Pro-Russia / Anti-Ukraine Tweets compared to Pro-Ukraine / Anti-Russia and Neutral Tweets. To account for this, synthetic Tweets were generated for the Pro-Russia / Anti-Ukraine classes using data augmentation methods. For each labeled Pro-Russia / Anti-Ukraine Tweet, a modified version of the Tweet was generated using the `ContextualWordEmbsAug` word augementer from the `nlpaug` library, which replaces word(s) in a body of text with a contextually relevant word[12]. This process resulted in an additional 161 Tweets that were used as training examples for the final iteration of the Neural Network.

## 4.2 Classification

### 4.2.1 Sentiment Analysis

Sentiment Analysis was performed using a VADER Sentiment Analysis model, which includes a lexicon specifically tuned to analyze text from social media[10]. The built-in VADER sentiment analyzer as part of the Python Natural Language Toolkit (`nltk`) was used. Raw text from Tweets was passed into `nltk.sentiment.vader.SentimentIntensityAnalyzer` to generate a compound score that indicated the sentiment as either positive, negative or neutral.

Results were written into 5 separate files depending on the category determined by the model, either Pro/Anti Russia/Ukraine, or Neutral [see sec 2]. Results were verified by taking the first 250 Tweets in each category when sorted from highest to lowest absolute value, labeling them, and comparing them to their expected results.

### 4.2.2 Clustering

k-means clustering was performed on 10,000 Tweets. Tweets were chosen as the Top 2,000 from each category that Tweets were binned in from Sentiment Analysis when sorted by absolute sentiment value. No specific methodology was used in selecting the 2,000 Neutral Tweets with a sentiment value of 0.

The built-in k-means clusterer as part of the Python Natural Language Toolkit (`nltk`) was

used. Distance from each Tweet to its assigned cluster centroid was measured using the cosine distance. Tweets were manually tokenized before analysis. This process included converting text to lowercase, stripping punctuation, URLs and stopwords, and stemming words using the `nltk.stem.SnowballStemmer`. Tweets were then converted into features utilizing a TF-IDF dense matrix, which considers the overall frequency of words throughout all Tweets[2].

Multiple cluster sizes were tested and then analyzed after performing PCA decomposition, which mapped the clusters into 2 dimensions. It was determined that a cluster size of 3 best fit the data, as when more clusters were added, cluster centroids began to overlap, making several clusters indistinct from each other. The magnitude of the individual features, i.e. words, was also measured to determine the most influential words on each cluster, where a feature with a larger magnitude was determined to have greater influence on the cluster. Once binned into 3 clusters, Tweets were sorted from smallest to largest distance from their cluster centroid. From each cluster, the 250 Tweets closest to the centroid were manually labeled to determine if any common category emerged from a cluster.

### 4.2.3 Classification Neural Network

Tweets were classified using a Neural Network based on PyTorch's sample code for text classification models, `text_sentiment_ngrams_tutorial.py`[21]. The Neural Network represented features as an embedding bag dense matrix, with dimensions of `vocab length x input layer size`. The network contained a single layer with a sized input and output that were dynamically tuned. `CrossEntropyLoss` was utilized as the loss function to compute the difference between the input and the target[4]. The optimizer was dynamically chosen, but all optimizers tested were variations of Stochastic Gradient Descent[19]. A scheduler was used to decay the learning rate based on the gamma value[5]. Tweet text was tokenized using `torchtext`'s built-in English language tokenizer along with n-grams with `n=2`[7].

The model used the following hyperparameters:

- Epoch - Number of training data runs through the model[16]. Kept constant at 25.
- Input Layer Size - Size of neural network input layer. Ranged from 16 to 32.
- Output Layer Size - Size of neural network output layer. Ranged from class size to 25.
- Learning Rate - Controls parameters' rate of change[16]. Ranged from 5e-5 to 5e2.
- Gamma - Rate of learning rate decay[6]. Ranged from 1e-3 to 1e-1.
- Optimizer - Algorithm for choosing ideal weights for neural network[1]. One of Adam, Adadelta, RMSprop, or SGD.

The values of these hyperparameters were chosen automatically using Optuna, a hyperparameter optimization framework[14]. Optuna ran the model through multiple trials using the holdout method with an 80/10/10 split between train, validate, and test datasets. This allowed Optuna to gradually alter the hyperparameters in order to choose the values that

resulted in the best performance on the model. Results were saved and analyzed using MLflow, a machine learning framework used for processing experimental results[13]. Once an ideal set of hyperparameters was chosen, the model was retrained and evaluated using k-fold cross-validation with k=5 and class stratification among folds.

The model was run for the following classification subsets:

- 3 total classes - Classification as either Pro-Ukraine, Pro-Russia, or Neutral. Anti-Russia Tweets were grouped with Pro-Ukraine, and vice versa. Tweets with other labels were discarded.
- 5 total classes - Classification as either Pro/Anti-Ukraine, Pro/Anti-Russia, or Neutral. Tweets with other labels were discarded.
- 7 total classes - Classification as per sec 4.1.1. No Tweets discarded.

Models were analyzed in terms of accuracy, where the model with the best performance was equivalent to the model with the highest measured accuracy. A weighted F1 score of the final model was also measured, which considers the precision and recall of the model in terms of the classes represented in the dataset. The model using 3 total classes was tested with and without the augmented dataset, whereas all other models were tested without it[sec 4.1.4].

## 5 Results

### 5.1 Classification via Sentiment Analysis

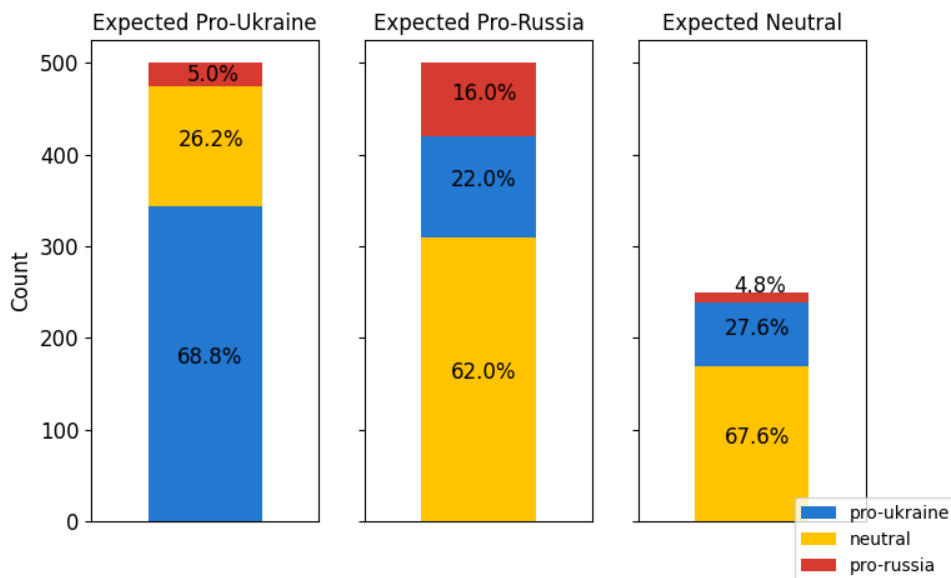


Figure 1: Tweets Labeled for each Expected Sentiment

## 5.2 Classification via k-means Clustering

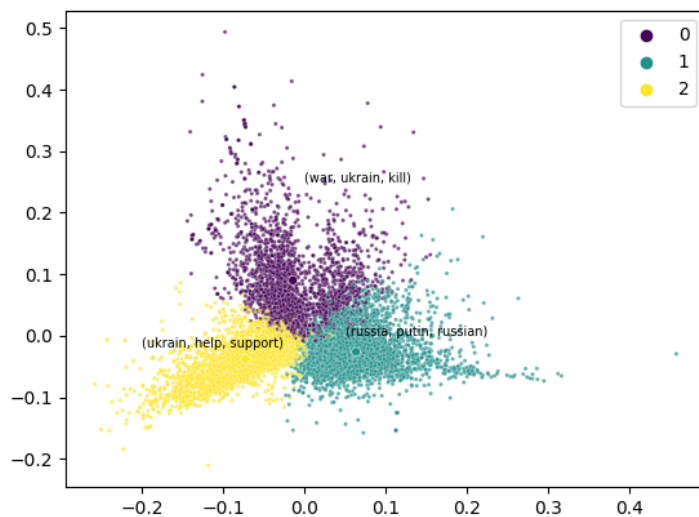


Figure 2: PCA Representation of k-means Clustering with k=3 and 3 Most Significant Features for each Cluster Listed

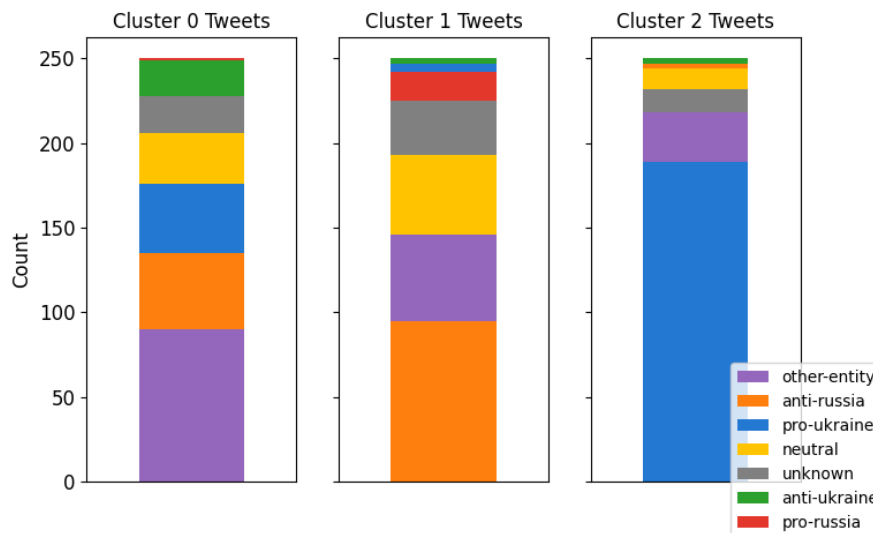


Figure 3: Tweets labeled as One of Any Possible Category for each Cluster

	Pro-Ukr	Pro-Rus	Anti-Ukr	Anti-Rus	Neutral	Other Entity	Unknown
0	41, 16.4%	1, 0.4%	21, 8.4%	45, 18.0%	30, 12.0%	90, 36.0%	22, 8.8%
1	5, 2.0%	17, 6.8%	3, 1.2%	95, 38.0%	47, 18.8%	51, 20.4%	32, 12.8%
2	189, 75.6%	0, 0.0%	3, 1.2%	3, 1.2%	12, 4.8%	29, 11.6%	14, 5.6%

Table 1: Tweets labeled for All Categories in each Cluster. Values are Count and Percent of Category in Cluster



### 5.3 Classification via Neural Network



Figure 4: Confusion Matrix of all 5-fold Cross-Validation Results from Neural Network using Pro-Ukraine, Pro-Russia, and Neutral Categories with Augmented Training Data

Data Type	Input Layer Size	Output Layer Size	Gamma	Learning Rate	Optimizer
Augmented	27	5	0.00393	0.08281	Adam
Not Augmented	21	21	0.01314	0.07355	RMSProp

Table 2: Hyperparameter Values for Model With and Without Augmented Data

Data Type	Pro-Ukraine	Pro-Russia	Neutral
Augmented	91.45%	32.1%	56.67%
Not Augmented	93.78%	8.02%	55.19%

Table 3: Individual Category Accuracy for Models With and Without Augmented Data

## 6 Discussion

Among the 3 methods analyzed, the Supervised Neural Network performed the best in labeling Tweets by their Desired Categories when compared to Sentiment Analysis and Clustering methods. Sentiment Analysis and Clustering were effective in grouping some related Tweets together and performed better than randomly assigning categories in most instances. However their performance was too inconsistent to be taken as absolute truth when attempting to assign categories to novel data. Although limited by the amount of labeled data, the Neural Network performed consistently well, with a 77% accuracy and weighted F1 score of 0.76 when categorizing Tweets as Pro-Ukraine, Pro-Russia, or Neutral.

When performing Sentiment Analysis, the hypothesis that Tweets with a positive sentiment towards a topic related to Russia or Ukraine would indicate support for those countries, and vice versa for a negative sentiment as stated in sec 2 was tested. This correlation performed decently well when analyzing Tweets that were expected to be in support of Ukraine and for Neutral Tweets. When categorizing Tweets as either Pro-Ukraine, Pro-Russia, or Neutral, 68.8% of Tweets expected to be Pro-Ukraine were actually Pro-Ukraine and 67.6% of Tweets expected to be Neutral were actually Neutral. However, Sentiment Analysis failed to generate a consistent method for identifying Pro-Russia Tweets. 16.0% of Tweets expected to be in favor of Russia actually did fall under this category, which was 3x as many found in other groups. However, there were actually more Pro-Ukraine Tweets (22.0%) in this group, and a majority of Tweets were Neutral (62.0%). While there was some correlation between sentiment and a Tweet’s stance on the conflict, it was not a reliable method for categorization.

Clustering was performed with the hypothesis that Tweets in the same cluster would fall under the same category, even if more or less clusters existed than the Desired Categories. k-means with a cluster size of 3 was used, as it was the only value of k to show distinct clusters when analyzing the data after performing Principal Component Analysis (PCA).

Upon examining the Tweets in each cluster, common themes did arise. Cluster 0 had more Tweets related to the conflict but criticizing other entities besides Russia or Ukraine, composing 36% of the labeled Tweets in Cluster 0. Cluster 1 had the most Anti-Russian Tweets, with 38% of the labeled Tweets being Anti-Russia. This is further evidenced by the 3 most significant features of the cluster: “Russia”, “Putin”, and “Russian”. Additionally, this cluster did not include many Pro-Ukraine Tweets, as only 5 Pro-Ukraine Tweets were found in the 250 Tweets labeled. Cluster 2 was the most consistent in terms of categorization, with 75.6% of the labeled Tweets being Pro-Ukraine. Many of the Tweets in this cluster referred to raising money and aid for Ukraine, evidenced by the cluster’s most significant features, “Ukrain”, “help”, and “support”.

Although Cluster 2 showed some consistency among categories, the Tweets labeled in each cluster were those intended to have the strongest correlation with its centroid. It is likely that as Tweets further from the centroid were analyzed, the relationship with the cluster would become weaker, making it ineffective to correlate a cluster with a specific category.

The Neural Network with the hyperparameters defined in Table 2 showed the most promise for accurately classifying Tweets. When simplifying the Desired Categories as either Pro-Ukraine, Pro-Russia, or Neutral, the Neural Network classified Tweets with a mean accuracy of 77% and a weighted F1 score of 0.76.

The Neural Network is not without issues. Because of the limited nature of the dataset, the model had particular trouble classifying Pro-Russian Tweets, scoring only 8% accuracy when training a model without the augmented data detailed in sec 4.1.4. Training with augmented data improved the accuracy nearly 4x, but the model still performed with only 32.1% accuracy among Pro-Russian Tweets. Categories with more data, such as the Pro-Ukraine Tweets, performed much better, as these Tweets were classified correctly 91% of the time. This success is indicative of the fact that Pro-Russian and Neutral Tweets have the potential to be more accurately classified by expanding the labeled dataset. This can be done by manually labeling more Tweets, or by further experimenting with data augmentation methods on the existing data.

Sentiment Analysis and Clustering proved useful in labeling Tweets, as grouping Tweets with similar themes together made it easier to manually label them. However, their performance on raw data was not sufficient for grouping Tweets with the same Desired Categories together. The Neural Network showed the most promise in accurately labeling Tweets, and would be the best model to use for accurate classification of novel data.

## 6.1 Future Work

The methods described in this paper could be expanded for further study. For Sentiment Analysis, an Aspect-Based Sentiment Analysis model which specifically focused on the desired aspect could be compared to the results achieved by using the VADER model[18]. This type of Sentiment Analysis model may do a better job of detecting the aspect a Tweet is referring to, and therefore the sentiment values may be more relevant. For clustering, different methods besides k-means could be evaluated. Additionally, a more formal method of determining cluster size besides analyzing a PCA plot could be implemented. For Neural Network Classification, the current model could be expanded to a deep network with more complex layerings. This would likely require more data, as only 2,000 labeled Tweets are currently available. The dataset could be expanded by collecting and labeling more data, or by using a 3rd party service to label a significantly larger batch of Tweets among all categories.

## 7 Acknowledgements

Dr. Zoran Obradovic and Marija Stanojevic's guidance through Temple's KDDM class made the educated use of these methods possible. The Python Foundation and the maintainers of the open-source libraries listed in the appendix allowed for several complex data mining methods to be quickly implemented. The Twitter Intelligence Tool, as well as the users whose Tweets were scraped using it, allowed for an abundance of data to be used for this project.

## References

- [1] *A comprehensive guide on Deep learning optimizers*. Oct. 2021. URL: <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/#:~:text=An%20optimizer%20is%20a%20function,loss%20and%20improve%20the%20accuracy..>
- [2] Mukesh Chaudhary. *TF-IDF vectorizer scikit-learn*. Jan. 2021. URL: <https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a>.
- [3] *Conflict in Ukraine*. URL: <https://www.cfr.org/global-conflict-tracker/conflict/conflict-ukraine>.
- [4] Torch Contributors. *Crossentropyloss*. 2019. URL: <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html#torch.nn.CrossEntropyLoss>.
- [5] Torch Contributors. *STEPLR*. 2019. URL: [https://pytorch.org/docs/stable/generated/torch.optim.lr\\_scheduler.StepLR.html](https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.StepLR.html).
- [6] Torch Contributors. *TORCH.OPTIM*. 2019. URL: <https://pytorch.org/docs/stable/optim.html>.
- [7] Torch Contributors. *TORCHTEXT.DATA.UTILS*. 2017. URL: [https://pytorch.org/text/stable/data\\_utils.html](https://pytorch.org/text/stable/data_utils.html).
- [8] IBM Cloud Education. *Supervised Learning*. Aug. 2020. URL: <https://www.ibm.com/cloud/learn/supervised-learning>.
- [9] Ibrahim R. Hallac, Betul Ay, and Galip Aydin. “Experiments on Fine Tuning Deep Learning Models With News Data For Tweet Classification”. In: *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. 2018, pp. 1–5. DOI: 10.1109/IDAP.2018.8620869.
- [10] C.J. Hutto and E.E. Gilbert. “VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text”. In: *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*. Ann Arbor, MI, June 2014. (2014). URL: <http://eegilbert.org/papers/icwsm14.vader.hutto.pdf>.
- [11] Abhinav Kumar, Jyoti Prakash Singh, and Sunil Saumya. “A Comparative Analysis of Machine Learning Techniques for Disaster-Related Tweet Classification”. In: *2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129)*. 2019, pp. 222–227. DOI: 10.1109/R10-HTC47129.2019.9042443.
- [12] Edward Ma. *NLP Augmentation*. <https://github.com/makcedward/nlpaug/>. 2019.
- [13] *mlflow*. URL: <https://www.mlflow.org/>.
- [14] *Optuna: A hyperparameter optimization framework*. URL: <https://optuna.readthedocs.io/en/stable/index.html>.
- [15] Francesco Poldi. *TWINT - Twitter Intelligence Tool*. Apr. 8, 2022. URL: <https://github.com/twintproject/twint>.

- [16] Pranoy Radhakrishnan. *What are hyperparameters? and how to tune the hyperparameters in a deep neural network?* Oct. 2017. URL: <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a>.
- [17] Oliver Roeder. *We Gave You 3 Million Russian Troll Tweets. Here's What You've Found So Far.* Aug. 2018. URL: <https://fivethirtyeight.com/features/what-you-found-in-3-million-russian-troll-tweets/>.
- [18] Rafał Rolczyński. *Do You Trust in Aspect-Based Sentiment Analysis? Testing and Explaining Model Behaviors.* Mar. 2021. URL: <https://rafalrolczynski.com/2021/03/07/aspect-based-sentiment-analysis/>.
- [19] Sebastian Ruder. *An overview of gradient descent optimization algorithms.* Mar. 2020. URL: <https://ruder.io/optimizing-gradient-descent/>.
- [20] Pang-Ning Tan et al. "Artificial Neural Network (ANN)". In: *Introduction to Data Mining*. 2nd ed. Pearson, 2019, p. 4.7.
- [21] *Text classification with the TORCHTEXT library.* 2021. URL: [https://pytorch.org/tutorials/beginner/text\\_sentiment\\_ngrams\\_tutorial.html](https://pytorch.org/tutorials/beginner/text_sentiment_ngrams_tutorial.html).
- [22] Peter D. Turney. "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Un-supervised Classification of Reviews". In: (2002). DOI: 10.48550/ARXIV.CS/0212032. URL: <https://arxiv.org/abs/cs/0212032>.

# Appendices

## A Code and Results

The code used in the implementation of the methods described in this paper, as well as experimental results, are located at:

<https://github.com/AGnias47/russo-ukranian-tweet-classification>.

## B Program Information

### B.1 System Specifications

The system used to run all methods described in this paper has the following specifications:

OS: Ubuntu 21.10

RAM: 16 GB

GPU: N/A

Python Version: 3.9.7

### B.2 Third Party Libraries

matplotlib==3.5.1

mlflow==1.25.0

nlpaug==1.1.10

nltk==3.7

numpy==1.22.3

optuna==2.10.0

pandas==1.4.2

pytest==7.1.1

scikit\_learn==1.0.2

torch==1.11.0+cu113

torchttext==0.12.0

tqdm==4.64.0

seaborn==0.11.2

sklearn==0.0

scikit-learn==1.0.2

# C Additional Results Data

## C.1 Sentiment Analysis

label	+ Ukraine	+ Russia	- Ukraine	- Russia	Neutral
Pro-Ukraine	189	1	32	0	68
Pro-Russia	0	56	2	21	2
Pro-Ukraine	0	1	21	4	10
Pro-Ruussia	1	56	21	154	1
Neutral	6	29	11	11	124
Other Entity	8	39	92	42	21
Unknown	46	68	71	18	24

Table 4: Tweets labeled for Desired Categories after performing Sentiment Analysis for 5 expected categories

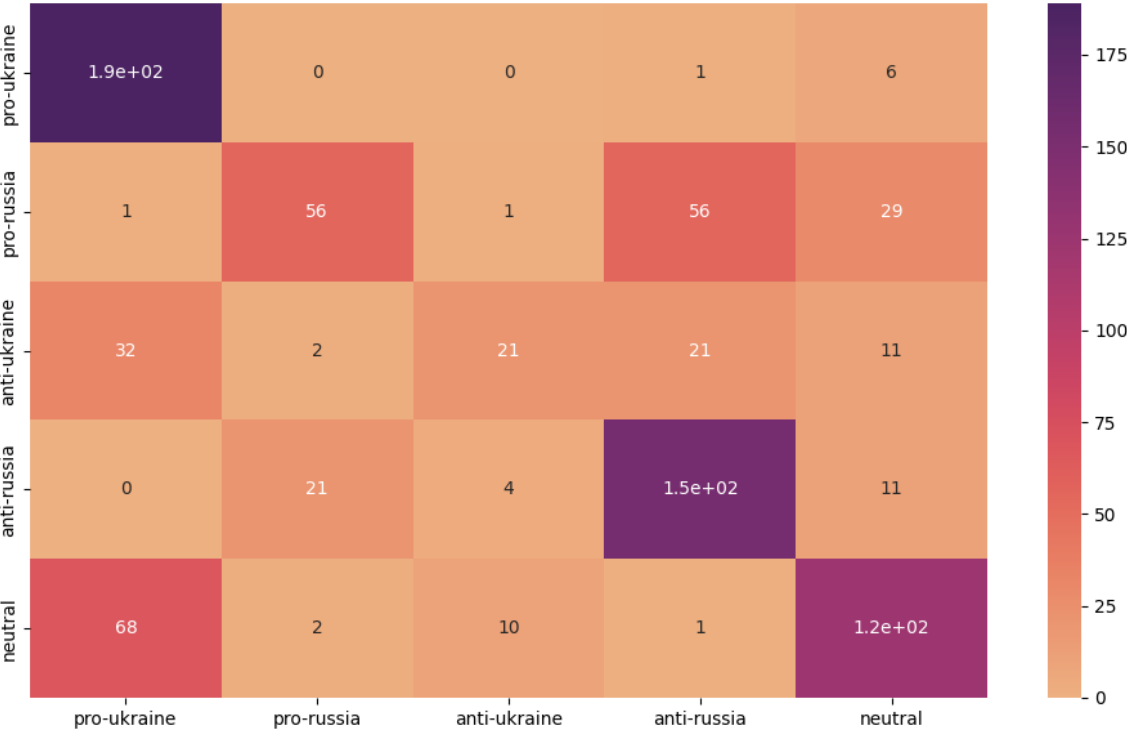


Figure 5: Confusion Matrix for Desired Categories after performing Sentiment Analysis for 5 expected categories

## C.2 Clustering

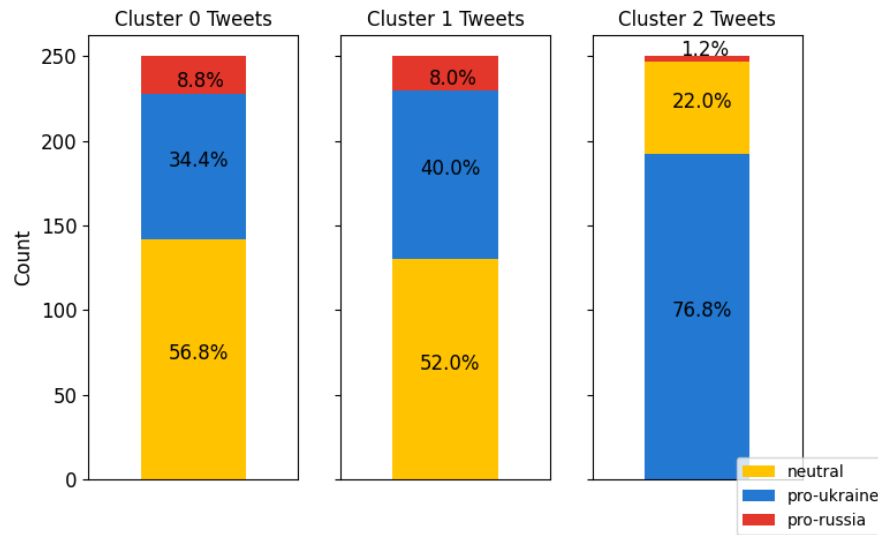


Figure 6: Tweets labeled as either Pro-Ukraine, Pro-Russia, or Neutral for each Cluster

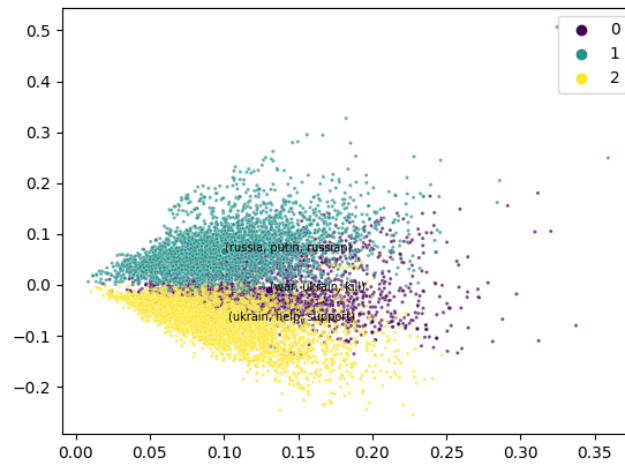


Figure 7: TruncatedSVD Representation of k-means Clustering with  $k=3$  and 3 Most Significant Features for each Cluster Listed



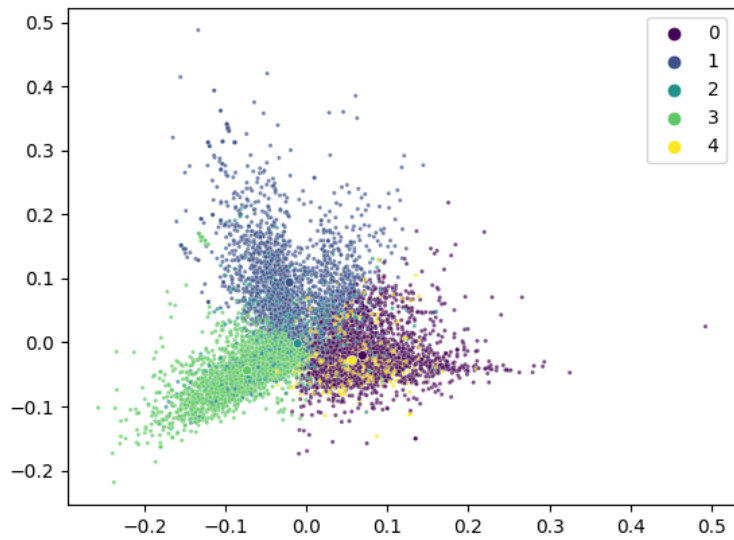


Figure 8: PCA Representation of k-means Clustering with k=5

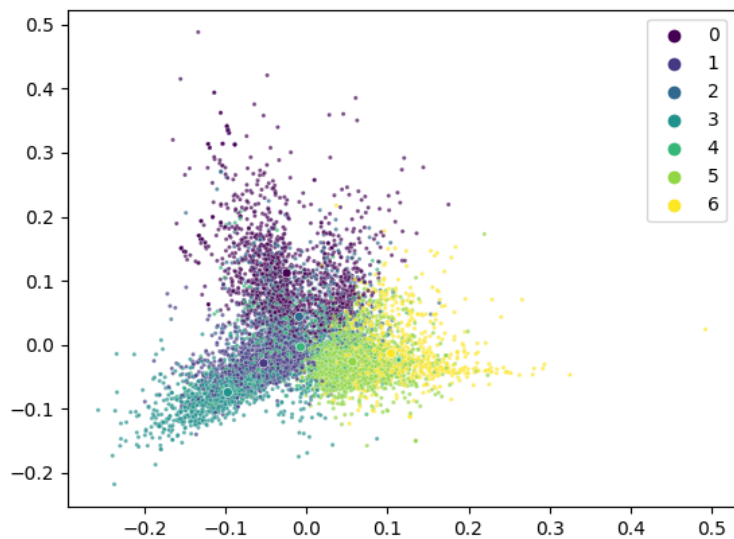


Figure 9: PCA Representation of k-means Clustering with k=7

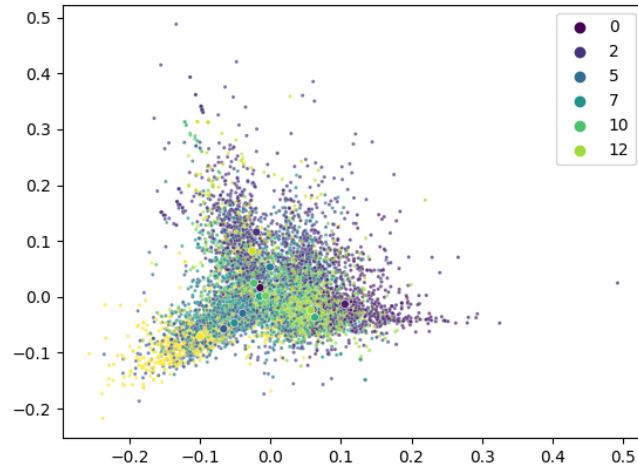


Figure 10: PCA Representation of k-means Clustering with  $k=15$

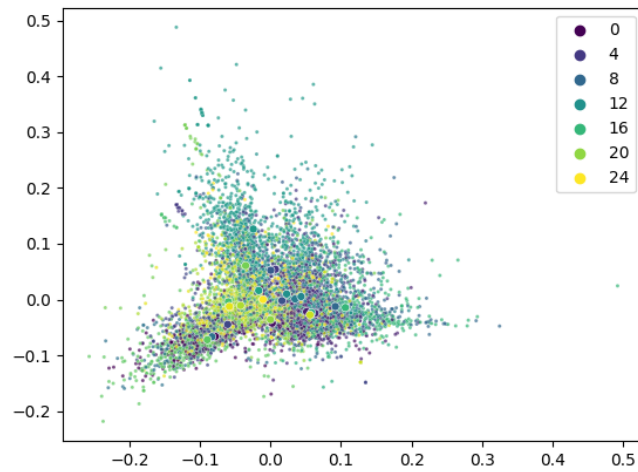


Figure 11: PCA Representation of k-means Clustering with  $k=25$

### C.3 Neural Network

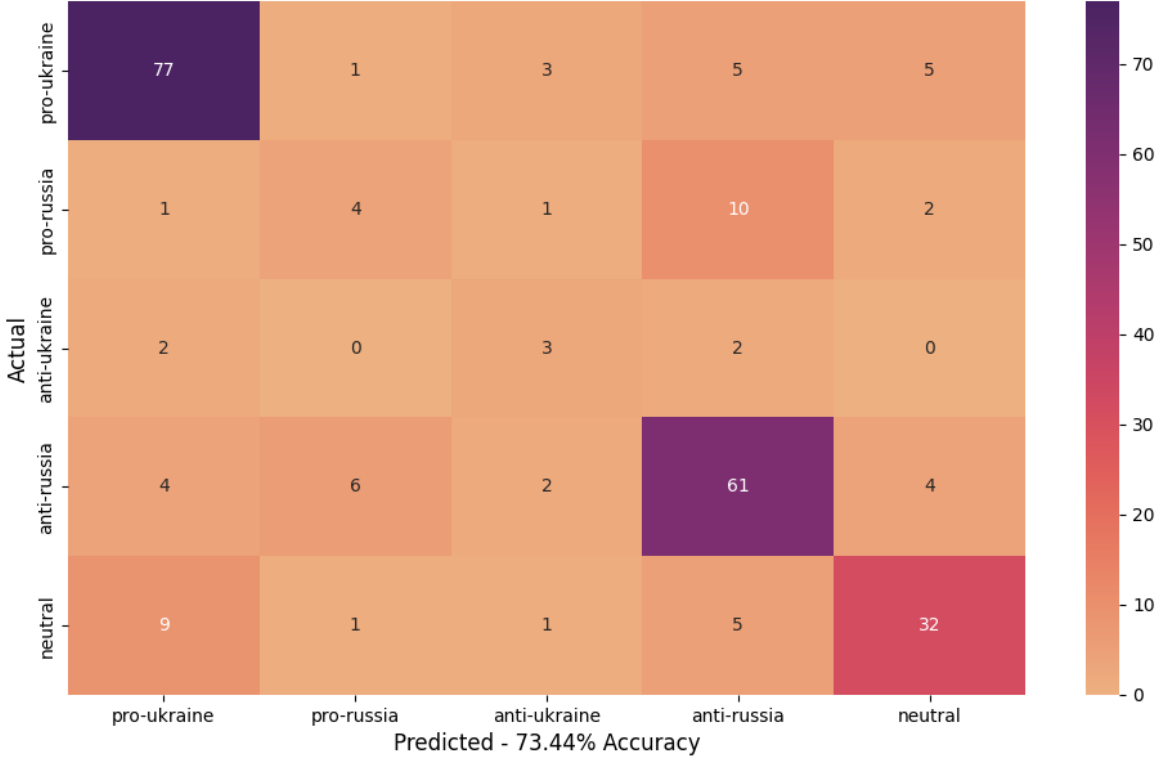


Figure 12: Confusion Matrix of results from a Neural Network classifying Tweets using Desired Categories. Holdout Method without Data Augmentation Used.

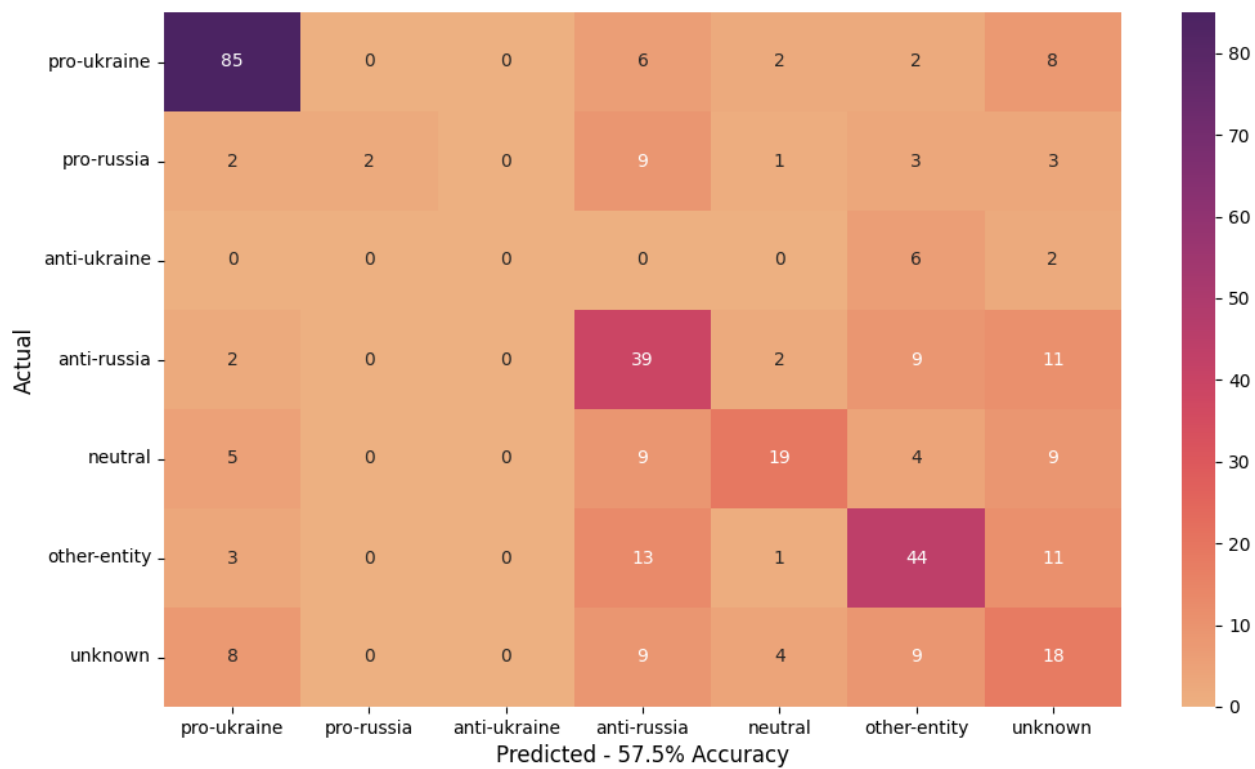


Figure 13: Confusion Matrix of results from a Neural Network classifying Tweets using All Possible Categories. Holdout Method without Data Augmentation Used.