Improving Emotion Detection through Translation of Text to ML Models Trained in Different Languages

Richard Hoehn*

Middle Tennessee State University

Dr. Jaishree Ranganathan†

Middle Tennessee State University

August 14, 2023

Abstract

This Qualifying Exam[1] research paper investigates the potential of improving Emotion Detection (ED) through translation of extended text data to multiple Machine Learning (ML) models trained in different languages. Our research focused on English (X[2] threads) and German (voting articles) datasets to improve prediction accuracy. By this our research aims to address the challenges posed by the lack of comprehensive labeled datasets and language fragmentation in ED research and projects.

By extending an original English language dataset with a translated dataset from German to English the training data (extended dataset) becomes larger and the hope is that better prediction rates can be achieved in ED analysis applications due to ML training having more data for learning. Furthermore, translating English to German to extend the German dataset and training another ML model using PySpark and accessing both models in real-time by a simple RESTful API may improve the prediction rates even more by dual processing a sentence at the same time.

In order to present the results presented both orally to the MTSU's Computational Data & Science Committee and within this paper labeled datasets both in English and German were collected, parsed, cleaned, translated, and two ML models built that can be accessed via an API to get a prediction by inputting sentences both in German or English via GET method using a simple query-string.

Sadly the research and results point to the realization that extending the datasets by translation did not improve prediction rates of English or German. Some minor gains can be achieved by accessing both English and German models at the same time via the RESTful API but unfortunately the benefits may not really support the efforts. In conclusion it that much more data needs to be collected in order to be able to definitely answer this research question.

*Electronic address: rhoehn@mtmail.mtsu.edu; corresponding author

†Electronic address: jaishree.ranganathan@mtsu.edu

[1]Comp. and Data Science PhD students are required to complete a Qualifying Exam in their first year

[2]X is formerly known as Twitter

Contents

1 Introduction

Emotion detection (ED) has gained significant interest and investment[15] in recent years as a means to understand human behavior and improve communication effectiveness. By analyzing text, speech or facial expressions, these detection processes try to accurately identify and classify emotions expressed by individuals or groups, often in real-time.

By employing a variety of ED processes, including natural language processing (NLP) and machine learning (ML), applications with the right amount of data can detect emotions with high precision[2]. These ED systems find applications in diverse fields, such as sentiment analysis, customer feedback analysis, voting and news stance prediction, and mental health monitoring, which all aim at contributing to a deeper understanding of human emotions and their impact on various aspects of life.

1.1 Human Emotion as a Theoretical Framework

The concept and number of emotions can vary depending on the theoretical framework or model being considered. One well-known model is the Plutchik's Wheel of Emotions seen in Figure 1, which proposes eight (8) primary emotions and their contrasting pairs. In this research paper we will be focusing on the basic eight (8) emotions and not on their sub-emotions or contrasting pairs. This allows the ML models being trained and built to be a multi-class classification model with eight labeled options.

The following are the eight primary emotions according to the Plutchik's Wheel of Emotions [19] that we will be focusing on both in our English and German datasets. A detailed reference of emotions used in this research can be found in Table 2 on page 10.

- Happiness: A feeling of Joy (Fun), contentment, or delight.
- Sadness: A state of unhappiness, sorrow, or grief.
- Anger: An intense emotional response associated with feelings of displeasure, frustration, or hostility.
- Worry: An emotional response triggered by perceived threats or danger, often associated with anxiety or terror.
- Surprise: A sudden and unexpected reaction to something unusual or startling.
- Disgust and/or Hate: A strong aversion or revulsion towards something unpleasant, offensive, or repulsive.
- Trust and/or Love: A sense of reliance, confidence, or faith in someone or something.
- Enthusiasm: A feeling of excited expectation or eagerness towards future events.

It's important to note that emotions are complex and multifaceted, and different emotion models and/or frameworks may propose alternative categorizations or variations in the number of primary

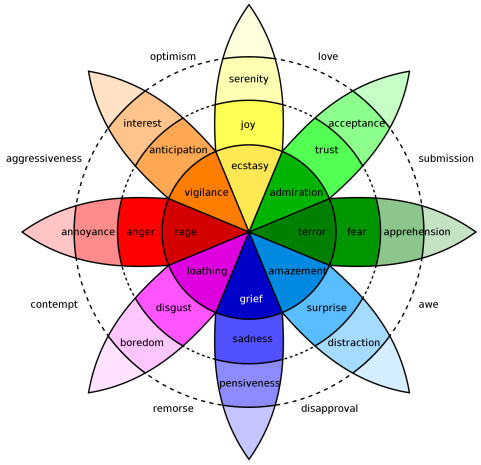

Figure 1: Plutchik Wheel of Emotions

emotions. However for the scope of this research project the most common eight (8) emotions are used as described above. This allows the ML models being trained and built to be a multi-class classification model with eight labeled classes.

By aligning the emotional labels across the English [6] and German [21] datasets that are being used in this research all eight listed above were linked to each other. Details on the linkage of the two datasets and their counts from data are described in Section 3.1 with an emphasis on Table 2's cross reference on page 10 of German and English language data.

1.2 Significance of Emotion Detection in Machine Learning

Emotion Detection (ED) analysis in Machine Learning (ML) models has gained significant attention due to its potential applications in various fields, including customer sentiment analysis, stance detection in regard to a specific targets such as news and voting[10], mental health monitoring[4], and human-computer interactions such as chat-bots[1] to just name a few.

By accurately identifying and understanding emotions from text data, ML applications can assist in improving user experiences, decision-making processes, and overall human-machine interactions in a positive manner[4, 10], with most of these interactions being processed in real-time.

As an example, the primary goals of chat-bots are entertainment, social contact, support, and novelty interaction, with a strong emphasis on productivity, scale-ability and cost reductions. In the case of chat-bots, there has been a immense requirement to simulate human-like characteristics and behaviors during human-machine interactions[8]. Per Kusal et al. [1] these chat-bots also improve customer engagement by offering friendliness, comforters, flexibility, and efficient assistance. As such chat-bots should provide customers with more engaging responses, directly addressing their issues or questions. The goal of deploying ED to these applications is for users perceive chat-bots as companions rather than simple assistants, which in turn helps the overall process of getting to a positive result. Per Adikari et al. the majority of user requests are emotional rather

than informative; therefore the need for chat-bots to gain the capacity to respond emotionally to customers via machine learning and sentiment analysis evolution[1] is very important and yields for many applications that use ED a way to improve interactions with machines.

> I usually know almost exactly how I feel. The problem is, I just can't tell anyone.
>
> – Meg Cabot

Due to the before mentioned wide scope of uses for ED this field of research and the work to advance it provides significance and importance in enabling machines to comprehend and respond appropriately to human interactions. Our research in turn contributes to the advancement of artificial intelligence, natural language processing (NLP) and in many cases the utilization of text-to-speech applications most all of us in today's society use and rely on [13] to go about our daily lives.

Lastly, in the paper by Chen et al.[2] based on emotion detection for online learning it states, interestingly, that with emotion recognition research, we need to focus on accuracy and real-time performance in order to apply ED based on physiological signals to solve practical problems; the key part of their claim is that in order for ED to be practical, real-time methods need to be used and deployed for maximum results.

1.3 Challenges: Lack of Labeled Datasets and Language Fragmentation

Emotion Detection is a fast growing field[17, 18], however unlike Sentiment Analysis (SA) the availability of large datasets for training purposes of ML models is much smaller [17, 13]. The primary reason of the lack of data stems from the face that emotion detection data requires primarily supervised learning data. In in turn requires time and effort to construct, catalog, and procure,; this all is mostly created and labeled by humans. To compound on the issue of the lack of labeled data; the many datasets that are available are in many cases in multiple languages - not all are in English since emotions that are linked to text are contextual in nature. for this research a a significant amount of time took place to find the German labeled data, which eventually was found online by google searching.

1.4 Motivation for the research project focusing on extending ED datasets through translation and evaluating the impact on prediction rates

This paper is significant because an entire industry of automated emotion reading (Text, Image, Video, and Sound) technologies is quickly emerging [17, 18]. The market for emotion recognition and detection software is forecast to reach at least $3.8 billion by 2025 [17, 18]. And since ED is already being incorporated into products for purposes such as marketing, robotics, driver safety, customer service, news and stance analysis, and a multiple of others, it is fitting to work on research to solve and / or improve issues in regards to prediction rates due to lack of data and language fragmentation.

With the before mentioned issues regarding the quantity (lack of) and language fragmentation of the ED datasets this research was motivated by finding a way to combine different language datasets into a single set for training processes to improve the predictability of ED. The aim of the research

is that by use of translation that the predictability of ED can be improved by extending an English dataset with translations from German and also process look-ups in real-time language conversion and processing.

1.5 Objectives & Scope of the Research

The primary objective of this Qualifying Exam research project is to address three (3) research questions, each aimed at improving the accuracy of predictive ED models in the context of English and German language datasets.

- The first hypothesis is that by introducing English data translated into German and extending an original German dataset, will the added text lead to an enhancement in its predictive capabilities. (Section 4.2)

- Similar to the first, can by translating German data to English and extending an original English dataset increase the predictability of English ED model? (Section 4.2)

- The third research question shifts the focus to real-time translation and its impact on prediction. Can by translating in real-time an input to multiple languages improve the predictability based on the combined output of two models that were trained in English and German to better the prediction results. (Section 4.3)

In summary, this research project centers around investigating innovative ways to enhance the predictability of Emotion Detection models in both English and German. With these three (3) objectives from above our methodology will be to procure, translate, train, and evaluate benefits of extending datasets by translation to improve emotion detection.

2 Literature Review

To ensure comprehensive coverage of the latest developments and research in emotion detection and analysis for this research paper, consideration and review was only given to incorporating research papers and online news articles that were published after the year 2015, thereby providing an up-to-date perspective on the subject matter at hand.

We believe by implementing the 2015 boundary, our methodology and results aim to provide a more up-to-date outlook on the subject matter at hand and what areas the emotion detection field is moving in with respect to machine learning research, applications and processes.

2.1 Emotion Detection Importance and Diverse Populations

Emotion Detection (ED) holds significant importance[15] in various domains, including psychology, social sciences, and human-computer interaction. By use of ML models[3] the analysis of human emotions at scale, providing valuable insights into individual and collective emotional states both in real-time but also as a type barometer for measuring sentiments from the past versus the current time. In Chowanda et al. paper[3] they believe that "Emotions hold a paramount role in the conversation, as it expresses context to the conversation.", this means that emotions are a part of a conversation and with that are needed to ensure valid analysis of a conversation.

Furthermore Kapoor et al. [7] states that, emotions can differ across age groups, genders, cultures, and languages. Including data from diverse populations helps in developing inclusive and culturally sensitive ED models. It ensures that the models are not biased towards specific groups and can accurately detect emotions in a wide range of individuals.

2.2 Lack of Labeled Datasets and Fragmentation caused by Different Languages

The lack of diverse and large-scale data hinders the development and training of accurate ED models, limiting their effectiveness and general ability across different contexts and populations. In recent research, efforts have been made to address these limitations by exploring new techniques for emotion detection.

For instance, a study by Kapoor[7] et al. proposes analyzing the entire spectrum of the language source to predict emotional changes. This provided some help, but quickly established itself as time consuming task and need of expertise in the subject matters that could not outweigh the results. Overcoming the scarcity of labeled datasets and embracing innovative approaches like the before mentioned study by Kapoor[7] et al. and their analysis can contribute to the advancement of ED research. Such advancements hold the potential to enhance the accuracy and applicability of emotion detection models, enabling a deeper understanding of human emotions and more effective responses in various domains, although the cost of such might be prohibitive at this time.

In addition, the paper by Mascarell[10] et al. states that unlike images language operate very differently, therefore the lack of labeled datasets poses a significant challenge in text-based Emotion Detection (ED) analysis as it limits the availability of reliable training data for ML models. This scarcity hampers the model's ability to learn and generalize emotions effectively.

And lastly, a paper by Kusal et al. further stipulates [9] that the fragmentation caused by different languages further exacerbates the issue, as it reduces the size and diversity of data available for training, resulting in limited cross-lingual generalization and potentially biased models.

Overcoming these challenges, lack of data and language fragmentation, is crucial for advancing emotion detection and analysis research for improving the accuracy, speed, and applicability of ML models across various languages and cultures.

3 Methodology

3.1 Data Procurement and Parsing of ED datasets in English and German

The data procurement was relatively straight forward and once found by using multiple Google search terms for the Emotion Detection in English and German text languages. Some of the main German data was obtained from the dataset built by ETH's Emotion and Stance Detection for German Text [10] which also hosts a good website with their findings[3]. The English dataset was downloaded from Kaggle[4] based on Tweets collected in 2021. The English dataset contains over 38,000 tweets and their corresponding emotion's labeled. Unfortunately the German dataset only contained just over 2,500 sentences and their respective emotions associated with it. Unfortunately this imbalance (see Table 1) of the two dataset sizes will had to be corrected and is explained in detail in the subsequent sections.

File Details		
Name	Row Count	Type
English	38,000	CSV
German	2,500	JSON

Table 1: File details of English & German data.

An important part of the process is to make sure we match / link the English emotions to those of the German language emotions. Since the two languages are dissimilar we opted to match exact emotions to each other that can be seen in Table 2 below. The linking yielded our eight (8) emotions we previously discussed in Sections 1.1. Details on how we formatted the the two (English and German) datasets to each other can be viewed in the Appendix A.2 and A.3.

The processing of two ED datasets to be synchronized or linked to the emotion is important. The first will be in English and the second in German. Each dataset will be translated into the other's language for extension purposes.

We decided to build the data to support that each row of each dataset will hold the English and German sentence. As an example Figure 2 depicts the original English dataset will be extended by the translated German (German $\Rightarrow$ English) dataset. By this simple process the English dataset got larger for training purposes.

The reason for selecting German in this research project was based on the author[5] being fluent in English and German and hence being able to easily traverse the two languages while working on the code for dataset translations and formatting. It was much easier to create the emotion cross reference Table 2 and also check the translation results due to the dual language benefit.

Once the two datasets have been translated, we now extend the original language dataset with the translated one. By this approach we are in essence extending the original datasets to create a larger one with the foreign language data. With both extended datasets we can now train our ML models

[3]https://mtc.ethz.ch/research/natural-language-processing/emotion-stance.html
[4]https://www.kaggle.com/datasets/pashupatigupta/emotion-detection-from-text
[5]Richard Hoehn

Emotion Datasets Labels				
English		German		Used
Name	Count	Name	Count	
Boredom	179	—	—	NO
Love	3842	Vertrauen	316	YES
Relief	1526	—	—	NO
Fun	1776	—	—	NO
Hate	1323	Ekel	29	YES
Neutral	8638	Unklar	314	YES
Anger	110	Ärger	226	YES
Happiness	5209	Freude	140	YES
Surprise	2187	Überraschung	369	YES
Sadness	5165	Traurigkeit	184	YES
Worry	8459	Angst	154	YES
Enthusiasm	759	Antizipation	774	YES
Empty	827	—	—	NO

Table 2: Dataset of English & German and their respective label counts.

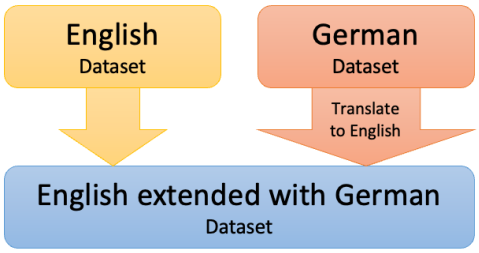

Figure 2: Example of English dataset extension by translating from German.

for English and German.

3.2 Translation Application for Converting English to German and Vice Versa

Translation of the emotion datasets was performed with Google Translator based on the Medium article[12] in which the author describes options for simple API translations. Google Cloud Translation was use mainly due to it's free nature and ease of use for this project. There are however many other text based API translation services that could have been used and might yield in some circumstance even better results.

As an example the following is a brief introduction to Google Translator's API with the code snippet below depicting the ease of use to translate text via an API.

```python
# Using Deep Translator to leverage Google Translate
# Link: https://cloud.google.com/translate/docs/reference/libraries/v2/python
from deep_translator import GoogleTranslator

sentence = 'Chocolate milk is so much better through a straw.'

translated = GoogleTranslator(source='auto', target='de').translate(sentence)
print(translated) # Schokoladenmilch schmeckt durch einen Strohhalm viel besser.

translated_back = GoogleTranslator(source='auto', target='en').translate(translated)
print(translated_back) # Chocolate milk tastes much better through a straw.
```

Source Code 1: Google Translate Example Code

Based on the above translation example the CSV[6] (English) and JSON[7] (German) datasets were individually loaded, parsed, and translated to their respective counter-part languages. Details of the application can be found in the Appendix A.2 for English and Appendix A.3 for the German processing. We decided to only use 1,500 rows from each language. This decision was made since the German (DE[8]) dataset was not much larger than that once filtering and processing was completed. This meant that the English (EN[9]) dataset was randomly paired down to 1,500 to match that of the German.

Each translated file now contains four (4) columns that are labeled in Tables 3 and 4 with the exact same column naming.

File: `pd_de_translated.csv`			
sentence_de	emotion_de	sentence_en	emotion_en
Original Sentence (feature)	Original Emotion (label)	Sentence Translated to English	Emotion set by Cross-Reference from German

Table 3: German CSV file structure after translation to English

File: `pd_en_translated.csv`			
sentence_en	emotion_en	sentence_de	emotion_de
Original Sentence (feature)	Original Emotion (label)	Sentence Translated to German	Emotion set by Cross-Reference from English

Table 4: English CSV File structure after translation to German

[6]CSV = Comma Separated Values
[7]JSON = JavaScript Object Notation
[8]DE = German
[9]EN = English

3.3 Training Multiple ML models with PySpark

After the completion of translating and processing the English and German datasets we come to training our models. Using the datasets from Tables 3, 4, and 5 that list an approximate ∼1,275 and ∼2,775 features for training.

It is important to note that in order to accurately test and compare the original and the extended datasets to each other we made sure that the test data was the same. to do this we randomly extracted 15% of the original datasets (English and German) before extending the original with the translated datasets.

```
# Split the English and German Dataframes for Training and Testing
# We are using a 15/85 Split
df_en_train, df_en_test = df_en.randomSplit([0.85, 0.15], seed=2023)
df_de_train, df_de_test = df_de.randomSplit([0.85, 0.15], seed=2023)

# Create the Extended Dataframe with Translated Data
# It is important to add the translated to the training data and not all of it
# Using *df_en_train.columns ensure that the columns are matched with the union
df_en_train_extended = df_en_train.union(df_de.select(*df_en_train.columns))
df_de_train_extended = df_de_train.union(df_en.select(*df_de_train.columns))
```

Source Code 2: Linking the English and German Emotions to the same label

In Chowanda et al. paper[3] on Text-based Emotion Techniques multiple algorithms were proposed, our research focused on just three (3) out of the six from the paper since the others had lower prediction rates based on the paperś conclusions. The algorithms tested in this research project ranged from the traditional machine learning to deep learning techniques, they are:

- **Naïve Bayes:** Per Pujan's paper[20] on the Naïve Bayes model, it classifies and/or assigns the most likely class to a given example based on its feature vector, simplifying the task greatly by assuming that the features are independent given a class.
- **Generalised Linear Model (GLM):** Per Mueller's paper on GLM[11] the Generalized linear models extend the concept of the well understood linear regression model.
- **Random Forest Classifier:** Per Schonlau's paper[16] on the Random Forest Classifier, they state that random decision forests easily adapt to non-linearity found in the data and therefore tend to predict better than linear regression. More specifically, ensemble learning algorithms like random forests are well suited for medium to large datasets.

Each of these algorithms used and their results are listed in the Results and Analysis Section 4 below.

The results of the testing resulted in some interesting parts that can be viewed in Table 6. We grouped the label by count from the test data (both English and German) and compared the predicted result counts to those. It became very clear that there was a strong bias on the GLM (Linear Regression) model that essentially proposed that all sentences evaluated to the same output both in English and German.

ML Models			
Model	Data	Rows for Training (85%)	Rows for Testing (15%)
A	English Original	1,275	225
B	English Extended By German	2,775	225 same as Model "A"
C	German Original	1,275	225
D	German Extended By English	2,775	225 same as Model "C"

Table 5: ML Models and their respective Train and Test Rows

3.4 Creating an API for Translation and Parallel Processing

In order for a user to access and get prediction results from all three (Random Forest, Naives Bayes, and Linear Regression) ML models trained in both English and German in real-time we crated a Jupyter Notebook that starts an Apache PySpark[10] session that is accessed via the local server listen on `https://127.0.0.1:8080`. The details of the RESTful the server can be found in Appendix B.2.

The server listens for GET requests and evaluates a querystring either called `sentence_en` or `sentence_de` if an English sentence is sent then the server translates to German and if a German is sent then it translates to English. Each sentence is then tokenized, formatted, and processed on the previously trained ML models in both languages.

Once all the the ML model predictions have completed, we then lookup the emotion associated with the individually predicted labels and build a response `JSON` object to pass back the calling application. The `JSON` object (Source Code: 3) contains the original sentence and translated, plus each emotions predicted label by the six previously trained models. In addition the main data, and `metadata` object is created that holds the PySpark version, start, and end times, including the processing time in seconds.

[10]PySpark: https://spark.apache.org/docs/latest/api/python/index.html

```
{
  "metadata": {
    "datetime": {
      "elapsed": 1.651038,
      "end": "Mon, 14 Aug 2023 12:10:35 GMT",
      "start": "Mon, 14 Aug 2023 12:10:33 GMT"
    },
    "spark": "3.3.2"
  },
  "predictions": {
    "de_lr": "enthusiasm",
    "de_nb": "relief",
    "de_rfc": "enthusiasm",
    "en_lr": "neutral",
    "en_nb": "love",
    "en_rfc": "happiness"
  },
  "sentence": {
    "english": "What a wonderful world this is!",
    "german": "Was für eine wundervolle Welt das ist!"
  }
}
```


Source Code 3: JSON output from API

3.5 Evaluation Metrics and Methodologies for Measuring the Performance

The evaluation of ML models trained on the English, German and their respective extension datasets is a critical step in assessing their performance and reliability. We chose to use the MulticlassClassificationEvaluator[11] because is a well known tool in the field of natural language processing (NLP) research, specifically in multi-class classification tasks such as emotion detection that has multiple labels associated with it; in our case eight (8) see Table 2 on Page 10. Generally this evaluator is used to evaluate the performance of machine learning (ML) models, using transformer-based architectures like BERT, RoBERTa, or DistilBERT[14], with DistilBERT[14] being the one we used in this research project.

The MulticlassClassificationEvaluator provides the essential evaluation metrics, including accuracy, Precision (eq: 2), Recall (eq: 4), and F1 Score (eq: 6) for ED classifications. These metrics provide valuable insights into a model's ability to correctly classify emotions across multiple classes and languages. Our research project is no different and the following measurements were used for our analysis:

- **Accuracy:** Accuracy measures the overall correctness of the emotion predictions made by the model. It is calculated as the ratio of correctly classified instances to the total number of instances in the dataset.

[11] URL: https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.evaluation.MulticlassClassificationEvaluator.html
[12] In the NB a result that cannot be matched is a zero label

ML Models Test versus Test Result Counts					
English					
ID	Label	Test Data	NB Res	GLM Res	RFC Res
0	boredom	0	25 [12]	0	0
1	love	26	8	3	11
2	Relief	0	33	0	0
3	Fun	0	5	0	0
4	hate	9	31	0	0
5	neutral	50	21	152	101
6	anger	0	36	0	0
7	happiness	39	44	5	17
8	surprise	10	9	0	1
9	sadness	29	0	2	19
10	worry	46	0	50	63
11	enthusiasm	3	0	0	0
German					
ID	Label	Test Data	NB Res	GLM Res	RFC Res
0	boredom	0	34	0	0
1	love	25	5	0	13
2	Relief	0	28	0	0
3	Fun	0	16	0	0
4	hate	5	16	0	2
5	neutral	30	24	0	5
6	anger	23	20	1	7
7	happiness	10	20	0	5
8	surprise	24	47	0	11
9	sadness	12	0	0	2
10	worry	9	0	209	4
11	enthusiasm	72	0	0	161

Table 6: Prediction Results Count compared to Test Data Label Counts

- **F1 Score:** Precision represents the proportion of correctly predicted emotions among all instances predicted as a specific emotion with respect to the labels. Recall measures the proportion of correctly predicted emotions among all instances that truly belong to a specific emotion. With these two the F1 Score ($F1_{Score} = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$) combines precision and recall into a single metric, providing a balanced evaluation.

While evaluating the models we focused on the Accuracy and the F1 Score to illustrate the prediction results of the original versus extended datasets, and also compared the different algorithms to each other. As an added example simply using the accuracy to evaluate a model can be miss-leading and therefore in our case using the F1 Score, that is directly correlated to the Precision (eq: 2 and Recall (eq: 4) is immensely important to understanding the ones models and finding the best of for

the task of Emotion Detection using Original and Extended datasets.

3.6 Using an API to Translate in Real-Time for Dual Prediction

The last stage of this research project was devoted to testing the API by use of a simple webpage. In order to accurately test and get real-world results and a sense on the real-time processing of a Emotion Detection a simple webpage was build to give a user the sense of the conclusion of the project. In Figure 3 a brief screenshot of some testing can be seen.

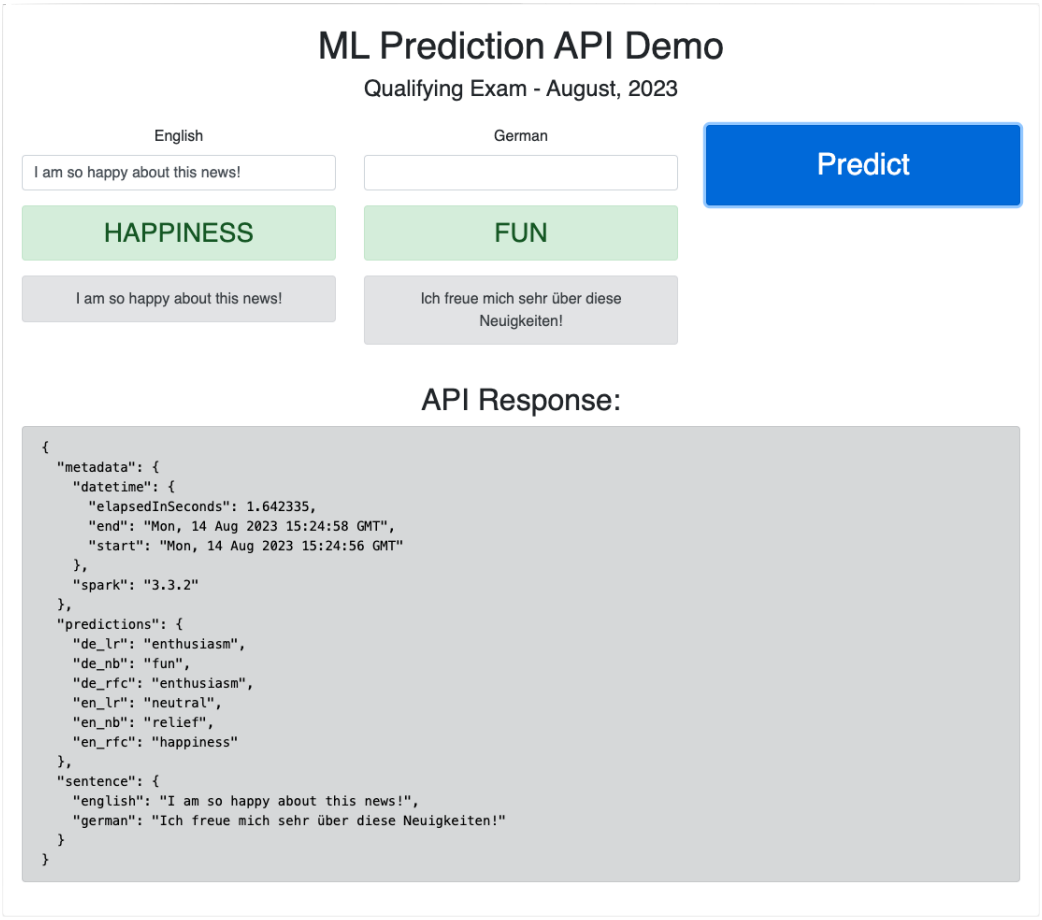

Figure 3: Demo API Webpage

In the screenshot (Figure: 3) example a sentence was entered in the English text-box and the "Predict" button pressed. The API took about ∼1.6seconds to translate the English to German and then process both sentences on the pre-trained ML models, build the `JSON` object and return it to the application.

3.7 Written & Oral Presentation of Research Finding

In conclusion of the project a research paper and oral presentation is required to pass the MTSU Computation Data Science PhD Qualifying Exam. The listing of all the datasets used, code

for the application, and details of the research project process in this research paper is openly available and hosted on a GitHub repository, allowing for easy access, reproducible, and collaborative engagement with the research findings. The repository is open and available under: `https://github.com/richardhoehn/mtsu.coms.qualifying-exam`[5]. In addition of all the code being listed in a public repository the paper was comprised and build using LaTeX[13].

Using LaTeX for this paper provides numerous benefits, some of which are portability, source control (due to plain-text nature), bibliography management, and last but not least the ability to write clean mathematical equations.

[13]LaTeX UR: https://www.latex-project.org/

4 Results and Analysis

The following section lists the prediction rates of training and testing of the original and extended datasets both in English and German. Overall results seem to indicate that there was no measurable benefit and even negative outcomes based on adding the extended data for Emotion Detection. In the below sub sections details on the results are listed by the common standards of using the below equations 1, 2, 4, and 6 for Accuracy, Precision, Recall, and the F1 Score. It should be noted that ED is a Multi-class classification problem, which means that the machine learning process assigns the predictions from the sentence data to one of several predefined emotion categories (see Table 2).

4.1 Prediction Results of Original & Extended Datasets

Overall the prediction results from the trained models is disappointing. Using the Random Forrest Classifier described in Section 3.5 provided the best prediction results based on a combination of the Accuracy and the F1 Score. It should be noted although the accuracy may be slightly better on the GLM models for both English and German, if one reviews the F1 Score, the RFC model performs better.

Oddly, the prediction results with the extended data where lower than using the original (non-extended) data for training. In the analysis and conclusions section 4.2 below details on why this may be the case are discussed in detail.

4.1.1 Formulas and Equations used in Analysis

The accuracy (eq: 1) of any machine learning model is a clear and easily comprehend-able way to understand the performance of a model. However, it is important to consider the context and potential trade-offs with other metrics, as high accuracy alone might not capture the full picture of a model's performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

The Precision (eq: 2) focuses in detail on the accuracy of positive predictions (TP)[14] made by the model. Per Google Cloud's definition the precision answers: What proportion of positive predictions where actually correct?

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Since our project is a multi-class classification project we will need to actually use Micro Averaging for the Precision. With this in mind we extend our simple equation 2 of precision to incorporate micro averaging is illustrated in equation 3.

$$Precision_{MicroAvg} = \frac{(TP_1 + TP_2 + \ldots + TP_n)}{(TP_1 + TP_2 + \ldots + TP_n + FP_1 + FP_2 + \ldots + FP_n)} \tag{3}$$

The Recall (eq: 4) focuses on the in detail on the accuracy of positive predictions[15] made by the

[14] Positive Predictions = TP (True Positive)
[15] Negative Predictions = FN (False Negative)

model. Per Google Cloud's definition: The recall attempts to answer: What proportion of actual positives.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

Since our project is a multi-class classification project we will need to actually use Micro Averaging for the Recall in a similar fashion to how we used it on Precision. With this in mind we extend our simple equation 4 of recall to incorporate micro averaging is illustrated in equation 5.

$$Recall_{MicroAvg} = \frac{(TP_1 + TP_2 + \ldots + TP_n)}{(TP_1 + TP_2 + \ldots + TP_n + FP_1 + FP_2 + \ldots + FP_n)} \tag{5}$$

The F1 score (eq: 6) in our project is important to assess the balance between precision and recall. In addition, the F1 score is especially relevant in multi-class classification scenarios where there are more than two classes.

$$F1_{Score} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6}$$

Lastly, we used micro averaging for the Recall, Precision, and F1 Score due to fact it give each label (Emotion see Table 2) equal weight regardless of the class label and the number of labels counted.

4.1.2 English Dataset Results

The Table 7 lists the results of all three ML models, their accuracy and F1 Score based on the English original and extended datasets. The improvements from the original and extended datasets are listed in the last column. In all cases the improvement to using the extended datasets was negative.

Results of English Dataset					
	Original Dataset		Extended Dataset		
	Accuracy	F1 Score	Accuracy	F1 Score	Improvement
Random Forrest	29.72%	25.83%	21.23%	19.43%	-28.56%
Naïve Bayes	5.66%	6.92%	3.77%	4.14%	-33.39%
GLM (lr)	32.08%	23.68%	25.47%	16.19%	-19.95%

Table 7: Model Testing Results on English Dataset

4.1.3 German Dataset Results

The Table 8 lists the results of all three ML models, their accuracy and F1 Score based on the German original and translated from English to German extended datasets. The improvements from the original and extended datasets are listed in the last column. Interestingly the Random Forrest Classifier was the only negative improvement whereas the GLM and Naive Bayes models

had improvements. However it should be noted that the prediction rates are so low in the GLM and Naive Bayes that they should not really be considered improvements.

Results of German Dataset					
	Original Dataset		Extended Dataset		
	Accuracy	F1 Score	Accuracy	F1 Score	Improvement
Random Forrest	28.10%	16.84%	25.71%	18.56%	-8.50%
Naïve Bayes	4.29%	4.31%	6.67%	4.21%	+55.48%
GLM (lr)	34.29%	17.57%	24.76%	18.27%	-27.79%

Table 8: Model Testing Results on German Dataset

The improvements on the accuracy are based on Equation (7) listed below. It is a simple and straightforward way of comparing the Accuracy from the original trained dataset to the extended trained dataset for purposes of comparing the potential benefits of extending the training datasets by translation.

$$Improvement(\%) = \frac{(Accuracy_{Ext}(\%) - Accuracy_{Org}(\%))}{Accuracy_{Org}(\%)} \tag{7}$$

It should be noted that although the improvement values are fairly large negatives those come from the fact that the accuracy is small and in most all cases the accuracy is around the 25 - 30% values and a change of just a few percent can have a significant impact on the improvement values.

4.2 Analysis of the impact of Extending Datasets by Translation

While extending datasets for machine learning training can often improve[15] model performance, it's important to recognize that this approach has its limitations, especially in ML models that can reach a limit on training data. In our case, relying on translation services for dataset expansion might not consistently yield better results when extending the data from translated text. These services, in our case using Google Translate[16] can introduce errors or inaccuracies in translations especially in the context of emotions of a particular text, leading to noisy or incorrect training data. Consequently, the models may learn from erroneous information, which did negatively impact our models performance on the ED tasks.

Specifically in our case of extending the original English and German datasets with translated data seems to indicate that the sentences and structure are not the same and therefore the prediction rates went down as the models got more diluted.

4.3 Impact Using API for Real-Time Translation

With the help of the webpage (Figure 3 on 16) a sense of the inaccuracy was also presented. Each round trip to the API took an between 1.4 and 1.6seconds. The prediction results mirrored those of Tables 7 and 8 which in turn made the usefulness of the translation not practical as can be seen in Table 9 below.

[16]Google Cloud Translate URL: https://cloud.google.com/translate

Webpage API Test					
	English		German		
Test	Emotion	Sentence	Emotion	Sentence	Time
A	Worry	This is bad idea and we need to stop right now	Anger	Dies ist eine schlechte Idee und wir müssen jetzt aufhören	1.49s
B	Happiness	What a wonderful world this is!	Relief	Was fur eine wundervolle Welt das ist!	1.65s
C	Happiness	I am so happy about this news!	Fun	Ich freue mich sehr uber diese Nauigkeiten!	1.64s
D	Sadness	Gosh I hate doing this today.	Relief	Meine Güte, ich hasse es heute.	2.56s

Table 9: API Webpage Test Results

4.4 Findings and Implications for Improving Emotion Detection

Based on the improvement results in Sections 4.1.2 and 4.1.3 it seems that using translation to extend both the English and German datasets resulted in a negative outcomes. The best outcomes were using a simple Linear Regression model on the English and German, however based on the F1 score (EQ 6) of the Random Forest Classifier it seems to perform the best on both English and German.

Unfortunately based on the results of extending the data our approach to also using real-time translation for multi-language models

4.5 Conclusion and Future Work

It's crucial to ensure the quality and accuracy of extended data, especially when language translation is involved, to prevent introducing more noise than meaningful information into the training process. Although the research did not garner any significant improvements by translating the German to English as a means on extending the English dataset, it actually made it the prediction results worse.

Future work includes testing and translating different models by using different translation services. Using Google Cloud's translation service was adequate for this research, however there are certainly better translation services that are available but are based on paid subscriptions.

References

[1] Achini Adikari et al. "A Cognitive Model for Emotion Awareness in Industrial Chatbots". In: *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*. Vol. 1. 2019, pp. 183–186. DOI: 10.1109/INDIN41052.2019.8972196.

[2] Mengnan Chen et al. "Research on Emotion Recognition for Online Learning in a Novel Computing Model". In: *Applied Sciences* 12.9 (2022). ISSN: 2076-3417. DOI: 10.3390/app12094236. URL: https://www.mdpi.com/2076-3417/12/9/4236.

[3] Andry Chowanda et al. "Exploring Text-based Emotions Recognition Machine Learning Techniques on Social Media Conversation". In: *Procedia Computer Science* 179 (2021). 5th International Conference on Computer Science and Computational Intelligence 2020, pp. 821–828. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2021.01.099. URL: https://www.sciencedirect.com/science/article/pii/S1877050921001320.

[4] Valentina Colonnello, Katia Mattarozzi, and Paolo M Russo. "Emotion recognition in medical students: effects of facial appearance and care schema activation". In: *Medical Education* 53.2 (2019), pp. 195–205. DOI: https://doi.org/10.1111/medu.13760. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/medu.13760. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/medu.13760.

[5] Richard Hoehn and Jaishree Ranganathan. *Improving Emotion Detection through Real-Time Translation of Text to ML Models Trained in Different Languages*. Version 1.0.1. 2023. URL: https://github.com/richardhoehn/mtsu.coms.qualifying-exam.

[6] Kaggle and data.world. *Emotion Detection from Text*. 2021. URL: https://www.kaggle.com/datasets/pashupatigupta/emotion-detection-from-text.

[7] Shalini Kapoor and Tarun Kumar. "Detecting emotion change instant in speech signal using spectral patterns in pitch coherent single frequency filtering spectrogram". In: *Expert Systems with Applications* 232 (2023), p. 120882. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.120882. URL: https://www.sciencedirect.com/science/article/pii/S0957417423013842.

[8] Sheetal Kusal et al. *A Review on Text-Based Emotion Detection – Techniques, Applications, Datasets, and Future Directions*. Apr. 2022.

[9] Sheetal Kusal et al. "AI Based Emotion Detection for Textual Big Data: Techniques and Contribution". In: *Big Data and Cognitive Computing* 5.3 (2021). ISSN: 2504-2289. DOI: 10.3390/bdcc5030043. URL: https://www.mdpi.com/2504-2289/5/3/43.

[10] Laura Mascarell et al. "Stance Detection in German News Articles". In: *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*. Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 66–77. DOI: 10.18653/v1/2021.fever-1.8. URL: https://aclanthology.org/2021.fever-1.8.

[11] Marlene Müller. "Generalized Linear Models". In: (Feb. 2004). DOI: 10.1007/978-3-642-21551-3_24.

[12] Nidhaloff Nidhaloff. *How to translate text with python*. July 2023. URL: https://medium.com/analytics-vidhya/how-to-translate-text-with-python-9d203139dcf5.

[13] Sajani Ranasinghe et al. "An Artificial Intelligence Framework for the Detection of Emotion Transitions in Telehealth Services". In: July 2022, pp. 1–5. DOI: 10.1109/HSI55341.2022.9869503.

[14] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *ArXiv* abs/1910.01108 (2019).

[15] Iqbal H. Sarker. "Machine Learning: Algorithms, Real-World Applications and Research Directions". In: *SN Computer Science* 2.3 (Mar. 2021), p. 160. ISSN: 2661-8907. DOI: 10.1007/s42979-021-00592-x. URL: https://doi.org/10.1007/s42979-021-00592-x.

[16] Matthias Schonlau and Rosie Yuyan Zou. "The random forest algorithm for statistical learning". In: *The Stata Journal* 20.1 (2020), pp. 3–29. DOI: 10.1177/1536867X20909688. eprint: https://doi.org/10.1177/1536867X20909688. URL: https://doi.org/10.1177/1536867X20909688.

[17] Jay Stanley. *"Experts Say 'Emotion Recognition' Lacks Scientific Foundation"*. 2019. URL: https://www.aclu.org/news/privacy-technology/experts-say-emotion-recognition-lacks-scientific.

[18] Jay Stanley. *"THE DAWN OF ROBOT SURVEILLANCE"*. 2019. URL: https://www.aclu.org/report/dawn-robot-surveillance.

[19] Erik Tromp and Mykola Pechenizkiy. "Rule-based Emotion Detection on Social Media: Putting Tweets on Plutchik's Wheel". In: (Dec. 2014).

[20] Pujan Ziaie et al. "A Naïve Bayes Classifier with Distance Weighting for Hand-Gesture Recognition". In: vol. 6. Jan. 2009, pp. 308–315. ISBN: 978-3-540-89984-6. DOI: 10.1007/978-3-540-89985-3_38.

[21] ETH Zurich. *CHeeSE Dataset*. 2019. URL: https://mtc.ethz.ch/publications/open-source/cheese-dataset.html.

List of Figures

A Appendix – Dataset Details

The following details in this appendix are for clarification purposes. All examples in this appendix are created via Panda and PySpark using data frames.

A.1 Translation Setup

All example code was created using Jupyter Notebooks.

```
import pandas as pd

# This was predtermined by review of the emotion from English to German
emotion_key = {
    "boredom": "---",
    "love": "Vertrauen",
    "relief": "---",
    "fun": "---",
    "hate": "Ekel",
    "neutral": "Unklar",
    "anger": "Ärger",
    "happiness": "Freude",
    "surprise": "Überraschung",
    "sadness": "Traurigkeit",
    "worry": "Angst",
    "enthusiasm": "Antizipation",
    "empty ": "---",
    "---": "Keine"
}

# Create a DataFrame from the emotion_key dictionary
df_emotions = pd.DataFrame(emotion_key.items(), columns=['emotion_en', 'emotion_de'])

# ********************
# *** Translation ***
# ********************
from googletrans import Translator
import time

# This google API take a Sentence and converts to German
def translate(sentence, dest_lang):
    try:
        translator = Translator()
        translator.raise_Exception = True
        translation = translator.translate(sentence, dest=dest_lang)
        time.sleep(0.5)  # Add a 1-second delay
        return translation.text
    except Exception as e:
        print(f"Translation Error: {e}")
        return None
```

A.2 Details on English Dataset Parsing and Formatting

The English dataset is based on tweets[17] that are saved as a csv[18] file. The following code was use to filter and process the file for use in this project.

```
# ********************************
# *** English Data Preparations ***
# ********************************
import pandas as pd
from tqdm import tqdm

# Load English CSV File into a Dataframe
df_en = pd.read_csv("data/english.csv")
print(f'Count (raw): {len(df_en)}')

# Print Columns
print(f'English Source Columns: {df_en.columns}\n')

# Filter only labels that we matched with the German counter part JSON File
filter_values = ["love", "hate", "neutral", "anger", "happiness", "surprise", "sadness", "worry", "enthu
df_en = df_en[df_en["sentiment"].isin(filter_values)]

# Find Empty Sentence Values
df_en = df_en[df_en["content"].notnull() & (df_en["content"].str.strip() != '')]

# Remove unnecessary column
df_en = df_en.drop("tweet_id", axis=1)

# Rename Columns
df_en = df_en.rename(columns = {'sentiment': 'emotion_en', 'content': 'sentence_en'})

# Merge German Emotions onto English
df_en = pd.merge(df_en, df_emotions, on='emotion_en', how='left')
df_en["sentence_de"] = "" # Add German Sentence Column
df_en = df_en.sample(n=1500, random_state=2023) # Randomly select 1500 rows

df_en.to_csv('./data/pd_en.csv', index=False)

# Iterate over the rows with tqdm to show the progress
for index, row in tqdm(df_en.iterrows(), total=df_en.shape[0]):
    # Call Translation
    sentence = translate(row["sentence_en"], 'de')
    # Save Sentence on Column
    df_en.at[index, 'sentence_de'] = sentence

df_en.to_csv('./data/pd_en_translated.csv', index=False)
```

[17] Tweets from Twitter
[18] csv = Comma Separated Values

A.3 Details on German Dataset Parsing and Formatting

The German dataset was downloaded from the ETH's[19] servers via "link" as a `JSON` file. Since the the file was `JSON` we will need to filter and explode the file based on teh emotions that are attached to it.

```
# *********************************
# *** German Data Preparations ***
# *********************************
import pandas as pd
from tqdm import tqdm

# Load German JSON File into a DataFrame
df_de = pd.read_json("data/german.json", lines=True)
print(f'Count (raw): {len(df_de)}')

# Print Columns
print(f'German Source Columns: {df_de.columns}\n')

# Explode the "article_emotion" column to distinct rows
df_de = df_de.explode('article_emotion')

# Filter only labels that we matched with the German counter part JSON File
filter_values = ["Vertrauen","Ekel", "Unklar", "Ärger", "Freude", "Überraschung", "Traurigkeit", "Angst
df_de = df_de[df_de["article_emotion"].isin(filter_values)]

# Find Empty Sentence Values
df_de = df_de[df_de["title"].notnull() & (df_de["title"].str.strip() != '')]

# Remove unnecessary columns
df_de = df_de.drop("article_id", axis=1)
df_de = df_de.drop("article_stance", axis=1)
df_de = df_de.drop("paragraphs", axis=1)
df_de = df_de.drop("source", axis=1)
df_de = df_de.drop("snippet", axis=1)

# Rename Columns
df_de = df_de.rename(columns={'title': 'sentence_de'})
df_de = df_de.rename(columns={'article_emotion': 'emotion_de'})

# Merge English Emotions onto German
df_de = pd.merge(df_de, df_emotions, on='emotion_de', how='left')
df_de["sentence_en"] = "" # Add English Sentence Column
df_de = df_de.sample(n=1500, random_state=2023) # Randomly select 1500 rows

df_de.to_csv('./data/pd_de.csv', index=False)

# Iterate over the rows with tqdm to show the progress
for index, row in tqdm(df_de.iterrows(), total=df_de.shape[0]):
```

[19] ETH (German: Eidgenössische Technische Hochschule) or Swiss Federal Institute of Technology

```
    # Call Translation
    sentence = translate(row["sentence_de"], 'en')
    # Save Sentence on Column
    df_de.at[index, 'sentence_en'] = sentence

df_de.to_csv('./data/pd_de_translated.csv', index=False)
```

B Appendix – Application & Code Details

In this appendix we aim to supply details to the code and application blocks of this project. It should be noted that some of the code has been edited to fit on these pages and might not represent verbatim the code in the GIT repository.

All code was developed and test using a MacBook PRO running `Ventura 13.4.1`. Since PySpark was developed for cluster processing it may seem counter intuitive to build PySpark code for a Mac; however the intent is that some of this research can be further used on clusters in the future.

B.1 PySpark Session Setup

```
# Setup Spark Session
from pyspark.sql import SparkSession

# Get Imports Needed
from pyspark.sql.functions import col, udf, regexp_replace, lower
from pyspark.sql import functions as F

# Get Datatypes needed for DataFrame manipulation
from pyspark.sql.types import IntegerType, StringType, ArrayType

# Other Imports
import re  # Import the "re" module for regular expressions

# Setup Spark Session
sc = SparkSession \
        .builder \
        .master("local[*]") \
        .appName("Qualyfing-Exam") \
        .getOrCreate()

# Print Spark Version being run
print("Spark V: ", sc.version)
```

B.2 RESTful API Server

The RESTful[20] API Server will be a simple Python Flask server listening on port `8080` on the local machine.

Flask is a lightweight and simple to use web framework in Python, allowing developers to create web applications and APIs. Due to it's simplicity and components design, Flask allowed us to rapidly build and interactive web service fro demonstration purposes on this project.

```
# Setup Spark Session
from pyspark.sql import SparkSession

# Import API Modules
from flask import Flask, request, jsonify
from flask_cors import CORS
import datetime

# Import Translator
from googletrans import Translator

# Impport Tokenizer and Pipeline Tools
from transformers import DistilBertTokenizer
from pyspark.ml.pipeline import PipelineModel

# Setup Spark Session
sc = SparkSession \
        .builder \
        .master("local[*]") \
        .appName("API-Demo") \
        .getOrCreate()

# Add Numerical Label for Emotions
emotion_key = {
    "boredom": 0,
    "love": 1,
    "relief": 2,
    "fun": 3,
    "hate": 4,
    "neutral": 5,
    "anger": 6,
    "happiness": 7,
    "surprise": 8,
    "sadness": 9,
    "worry": 10,
    "enthusiasm": 11,
    "empty": 12,
    "---": 13
}

```

[20]RESTful API is an interface that two computer systems use to exchange information securely over the internet.

```
# Initialize the AutoTokenizer
tokenizer = DistilBertTokenizer.from_pretrained("distilbert-base-multilingual-cased")

# Load Saved Models
model_en_rfc = PipelineModel.load("./models/model_en_rfc.model")
model_en_nb = PipelineModel.load("./models/model_en_nb.model")
model_en_lr = PipelineModel.load("./models/model_en_lr.model")
model_de_rfc = PipelineModel.load("./models/model_de_rfc.model")
model_de_nb = PipelineModel.load("./models/model_de_nb.model")
model_de_lr = PipelineModel.load("./models/model_de_lr.model")


app = Flask(__name__)
CORS(app)  # Use the CORS class to enable CORS for the entire app
translator = Translator()

# Setup Emotion Key Reverse
emotions = {v: k for k, v in emotion_key.items()}

@app.route('/predict', methods=['GET'])
def transform_data():
    start_time = datetime.datetime.now()
    sentence_en = request.args.get('sentence_en')
    sentence_de = request.args.get('sentence_de')

    print(f"English: {sentence_en}")
    print(f"German: {sentence_de}")

    if sentence_en:
        sentence_de = translator.translate(sentence_en, dest="de").text
    elif sentence_de:
        sentence_en = translator.translate(sentence_de, dest="en").text
    else:
        return jsonify({'error': 'No Sentence provided!'})

    print(f"English: {sentence_en}")
    print(f"German: {sentence_de}")

    words_en = tokenizer.tokenize(sentence_en)
    words_de = tokenizer.tokenize(sentence_de)

    # Create Englsih & German Dataframe
    df_en_api = sc.createDataFrame([(sentence_en, words_en)], ["sentence", "words"])
    df_de_api = sc.createDataFrame([(sentence_de, words_de)], ["sentence", "words"])

    predictions_en_rfc = model_en_rfc.transform(df_en_api)
    predictions_en_nb = model_en_nb.transform(df_en_api)
    predictions_en_lr = model_en_lr.transform(df_en_api)
    predictions_de_rfc = model_de_rfc.transform(df_de_api)
    predictions_de_nb = model_de_nb.transform(df_de_api)
    predictions_de_lr = model_de_lr.transform(df_de_api)
```

```
92
93      # Get End Time
94      end_time = datetime.datetime.now()
95      elapsed_time = (end_time - start_time).total_seconds()
96
97      #Build Response
98      resp = {
99          "sentence": {
100             "english": sentence_en,
101             "german": sentence_de,
102         },
103         "predictions": {
104             "en_rfc": emotions.get(predictions_en_rfc.collect()[0]["prediction"]),
105             "en_nb":  emotions.get(predictions_en_nb.collect()[0]["prediction"]),
106             "en_lr":  emotions.get(predictions_en_lr.collect()[0]["prediction"]),
107             "de_rfc": emotions.get(predictions_de_rfc.collect()[0]["prediction"]),
108             "de_nb":  emotions.get(predictions_de_nb.collect()[0]["prediction"]),
109             "de_lr":  emotions.get(predictions_de_lr.collect()[0]["prediction"]),
110         },
111         "metadata": {
112             "spark": sc.version,
113             "datetime": {
114                 "start": start_time,
115                 "end": end_time,
116                 "elapsedInSeconds": elapsed_time,
117             }
118         }
119     }
120
121     return jsonify(resp)
122
123 if __name__ == '__main__':
124     app.run(host='127.0.0.1', port=8080)
125
```