```
# Install the necessary libraries if they weren't already installed
!pip install pandas

# Install necessary libraries:
import pandas as pd
!python --version
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Python 3.10.12
```

What is the sentiment of the reviews, sorted by language?

We selected the Amazon Reviews Multilingual Dataset as our dataset: https://www.kaggle.com/datasets/mexwell/amazon-reviews-multi/data. It depicts a comprehensive collection of multilingual product reviews in CSV format. This dataset contains about 1.3 million samples in 6 languages (DE = German, EN = English, ES = Spanish, FR = French, JA = Japanese, ZH = Chinese) with the following features:

- **review_id**: A string identifier of the review.
- **product_id**: A string identifier of the product being reviewed.
- **reviewer_id**: A string identifier of the reviewer.
- **stars**: An int between 1-5 indicating the number of stars.
- **review_body**: The text body of the review.
- **review_title**: The text title of the review.
- **language**: The string identifier of the review language.
- **product_category**: String representation of the product's category.

The data will be directly downloaded and processed from the CSV files provided by the dataset. During preprocessing we will focus on removing stop words as well as choosing the correct embedding for the reviews.

7) Two new Columns:

- true_label
- predicted_label

In step 4, we split train.csv that contained the initial columns:

```
review_id,product_id,reviewer_id,stars,review_body,review_title,language,product_category.
```

Based on the "language" (de, en, es, fr, ja, zh) of each review, we divided them into 6 seperate files so that they are seperated by language for easier ressource handling. There, we added 2 more columns for true_label and predicted_label (see step 5+6 for sentiment analysis and accuracy). Therefore, this is how a row looks like in each of the reviews_language.csv files:

```
review_id,product_id,reviewer_id,stars,review_body,review_title,language,product_category,true_label,predicted_label.
```

What we want to do now: take the last two columns (true_label, predicted_label) of the 6 new files and add them back to the initial big dataset (train.csv). The complete file will then contain all languages again with the two new columns additionally.

We can match the rows (where each new value of true_label, predicted_label belongs to) via "review_id". Since they indicate the language and contain a number, this should be the easiest way to match the right values to the correct reviews. Example:

```
review_id,product_id,reviewer_id,stars,review_body,review_title,language,product_category,true_label,predicted_label.

de_0232738,product_de_0901865,reviewer_de_0033281,1,Kaum Saat aufgegangen.,Schlechte Mischung,de,lawn_and_garden,0,0
```

Get the initial dataset:

```
# Read csv-file:
file_path = 'train.csv'
train_df = pd.read_csv(file_path) # dataframe

# columns: id, review_id, product_id, reviewer_id, stars, review_body, review_title, language, product_category
```

Merge files:

The csv files contain the two new columns.

```
file_paths_languages = [
    'german_reviews_with_sentiments_and_accuracy.csv',
    'english_reviews_with_sentiments_and_accuracy.csv',
    'spanish_reviews_with_sentiments_and_accuracy.csv',
    'french_reviews_with_sentiments_and_accuracy.csv',
    'japanese_reviews_with_sentiments_and_accuracy.csv',
    'chinese_reviews_with_sentiments_and_accuracy.csv'
]

# init empty list
dataframes = []

# append the files to the list
for file_path in file_paths_languages:
    dataframes.append(pd.read_csv(file_path))

# concatenate all to one
combined_df = pd.concat(dataframes, ignore_index=True)

# save combined dataframe
output_file_path = 'train_with_sentiments_and_accuracy.csv'
combined_df.to_csv(output_file_path, index=False)

print(f"Combined data saved to {output_file_path}")
```

```
Combined data saved to train_with_sentiments_and_accuracy.csv
```