Text-to-Emoji Translator

A functional text-to-emoji translator capable of interpreting
and representing text with emoji sequences.

EMOJINET

https://www.kaggle.com/datasets/emirhanai/social-media-usage-and-emotional-well-being/data

Project Summary

Text-to-Emoji Translator

- **Our group created a neural network that translates phrases or sentences into sequences of emojis.**
- **The model identifies keywords and contextual meanings from the input text and maps them to the most relevant emojis.**

Agenda for today’s project review

- Discuss data extraction and cleaning techniques
- Present the Model, training and evaluation
- Review changes we made to the model
- Show overall model performance
- Discuss observations and what we’d do differently

Data

- **Train model with key-value (emoji) pairs**
- **Focus on base emojis (ignore modifiers and complex emojis)**
- **Augment data set to improve model predictions**

Step	1 – EmojiNet (Kaggle)	2 – Full Emoji List, V16.0	3 - Python emoji library
Collect	Machine-readable dictionary Upload, unzip, JSON df	Unicode.org website BeautifulSoup to parse tables	Extract ‘label’ & ‘emoji’ from emoji.EMOJI_DATA
Explore	Filtered to ‘name’ & ‘unicode’ Filtered to base emojis	Filtered to ‘label’ & ‘CLDR short name’	Filtered out unnecessary features
Clean	Remove null values Remove special chars Replace _ with a space Remove emoji modifiers	Removed null values Removed special chars Removed incomplete data	Remove null values Remove special chars Replace _ with a space Remove emoji modifiers
Integrate	Concatenate the three data sets in to combined_emoji_df[‘label’][‘emoji’] Oversample the data to improve model predictions Augmented data using ‘wordnet’ to identify label synonyms to improve model predictions (Natural Language Toolkit)		
Reduce	Loop through combined df and remove rows with duplicate label-emoji pairs		
No. Records	2,389	1,865	3,790

Understanding emoji Data

Base Emojis	Simple, standalone characters such as 😊 (Smiling Face), ❤️ (Red Heart), and 🌈 (Rainbow).
Complex Emojis	Combining base emojis to create more complicated characters, e.g., 👫 (People Holding Hands).
Modifiers	Used to alter base emojis, e.g., 👨 (Man with a White Beard). (U+200D Zero Width Joiner (ZWJ)
Unicode	Standardized system for encoding emojis. Each emoji is assigned a unique code point, e.g., U+1F601 for 😁 (Beaming Face with Smiling Eyes)

Overview of T5 Integration

- **Model Purpose:**
  - The T5 model (Text-to-Text Transfer Transformer) was utilized to map text inputs (e.g., "happy") to their corresponding emoji outputs (e.g., 😊).

- **Dataset Preparation:**
  - Combined data from multiple sources (EmojiNet, Unicode, and the Python Emoji library).
  - Preprocessed data included cleaning, oversampling, and augmenting labels with synonyms using WordNet to improve diversity.

- **Tokenization:**
  - Input texts (e.g., "happy") and target emojis were tokenized using T5 Tokenizer.
  - Applied padding and truncation for consistent sequence length.

Overview of T5 Integration

- **Training Process:**
  - Fine-tuned the "t5-small" model on a DataLoader with batch size = 16.
  - Optimized using AdamW with learning rate adjustments and weight decay.
  - Ran for 10 epochs, tracking and minimizing loss.

- **Inference:**
  - Model set to evaluation mode for translation.
  - Input phrases were tokenized and passed through the model to generate emoji outputs using beam search (num_beams=8).

- **Fallback Mechanism:**
  - If the model failed, a direct lookup in the emoji dataset was performed, ensuring robust predictions.

Learning by Experimenting

After 20 experiments:

- Enriching, augmenting data
- Using different models: t5-small, t5-base, & Bert
- Running up to 10 epochs
- Tweaking optimizer, batch, and other model settings

T5-small @ 10 epochs yielded the lowest loss

- Epoch 1, Loss: 513.9914644835517
- Epoch 2, Loss: 59.7617578310892
- Epoch 3, Loss: 35.85317394929007
- Epoch 4, Loss: 22.778702536423225
- Epoch 5, Loss: 14.844095607390045
- Epoch 6, Loss: 9.693488336226437
- Epoch 7, Loss: 6.586130286690604
- Epoch 8, Loss: 4.42386278442973
- Epoch 9, Loss: 3.8279994039039593
- Epoch 10, Loss: 2.760537032425418

Lessons Learned

- **Training with key-value pairs left predictive gaps:**
  - Lookup words are contained in multiple labels, e.g., apple vs. pineapple
  - Using base emojis alone to train the model—resulted in emojis not found

- **Hypothesis: A phrase to emojis data mapping data set would have better trained the model, improved context and predictions.**
  - For example: KomeijiForce/Text2Emoji · Datasets at Hugging Face

Being a nurse is a rollercoaster of emotions, from comforting patients to dealing with medical emergencies.		career
Can't wait to finally see my best friend tomorrow, I have missed them so much!		feeling
Pure bliss! Spend an entire day doing what you love can light up your soul like nothing else		feeling
Cruising along coastal highways in perfect harmony with nature on a motorcycle!		vehicle

References

- **EmojiNet**
  - https://www.kaggle.com/datasets/rtatman/emojinet?select=emojis.json
  - License: CC BY-NC-SA 4.0

- **Full Emoji List, v16.0**
  - All copyrights, trademarks and/or service marks associated with the emoji designs appearing on this website are the property of their respective owners. Any use of such copyrights, trademarks or service marks, including the reproduction, modification, distribution or republication of same without the prior written permission of the owner, is strictly prohibited.
  - Images in the charts are courtesy of Adobe, Apple, Emojipedia.org, EmojiXpress, Google, iDiversicons, Microsoft, and others.
  - Full Emoji List, v16.0

- **Emoji Python Library v2.14.0**
  - emoji — emoji documentation

Questions