EmojiSense: An Age-Aware Emoji Translator

Pinru Wang
pinruwan@usc.edu

Yihan Yang
yangyiha@usc.edu

Abstract

Emojis have evolved into a complex language with context-dependent meanings and distinct generational styles, yet current translation models often fail to capture these nuances. In this paper, we present EMOJISENSE, an age-aware translation model that interprets emoji sequences into English while adapting to specific age-based personas, including Gen Z and Boomers. We fine-tuned a BART-Large model on a diverse dataset of emoji-to-text examples, specifically synthesized to handle mixed contexts and distinct communication styles across different age groups. Our results demonstrate that this approach significantly improves both semantic accuracy (BERTScore) and stylistic fluency compared to non-contextual baseline models, confirming that demographic awareness is essential for accurate emoji interpretation.

1 Introduction

Emojis have transformed online communication, evolving from simple pictures into a complex language capable of expressing stories, emotions, and hidden meanings. Despite this depth, most past research on emoji-to-text translation relies on simple "pattern-based" or "rule-based" methods. These traditional approaches treat emojis as fixed dictionary symbols, failing to account for how meanings shift based on the context or the user's identity.

In reality, the meaning of an emoji is far from fixed. The same symbol can mean completely different things when used by different age groups. For instance, the skull emoji is usually interpreted by older generations literally, meaning danger or death. In contrast, younger users often use it to mean extreme laughter ("I'm dead").

These differences create a genuine barrier to communication. When users from different age groups interact online, conflicting emoji definitions

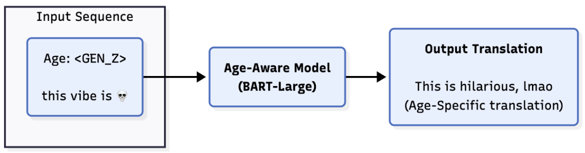


Figure 1: Expected workflow of our age-aware model

can lead to misinterpretations, awkward replies, and confusion. As emojis become increasingly central to daily messaging, this "generation gap" makes messages difficult to understand for readers outside a specific age group.

To address this, we argue that a good emoji translator must understand context and the user's age. In this paper, we introduce EMOJISENSE, a translation framework designed to capture these subtle differences. Our method follows a three-step training process, where we progressively improve the BART-Large transformer encoder-decoder model to handle more complex tasks.

The **first step** establishes a baseline by training the model to translate emoji-only sequences into English text. This serves as our foundation, learning the basic meaning of emojis without any extra context.

The **second step** advances this by training a context-aware model capable of handling mixed inputs (combinations of text and mixed-context emojis). By treating the surrounding text as hints, this model learns to clarify emoji meanings based on the sentence structure.

The **third step** introduces our novel age-aware mechanism. We train the model to accept an explicit "age parameter" as a control token, guiding the translation to reflect specific Gen Z or Boomer interpretations.

Our key hypothesis is that the fully context-and-age-aware model (Step 3) will significantly outperform the standard baseline (Step 1). We believe that by learning the specific communication styles of different generations, our model can achieve higher accuracy and fluency, correctly interpreting ambiguous emoji sequences that standard models would miss.

2 Related Work

Rule-Based Methods Early attempts at emoji translation relied mainly on simple rules. For example, in the study *An Approach for Text-to-Emoji Translation*, Wicke and Cunha built a system that translates text to emojis using a manual dictionary of over 400 "action-to-emoji" pairs (Wicke and Cunha, 2020). They describe their system as being "built around a dictionary," which serves as a classic rule-based baseline. While this approach works for basic tasks, it cannot handle complex meanings compared to modern machine learning methods.

Neural Network Translation Recent research has shifted towards neural networks, treating emoji-to-text translation as a sequence-to-sequence task similar to machine translation. *EmojiLM* (Peng et al., 2023) introduced a large, synthetically generated English-Emoji parallel dataset and demonstrated the effectiveness of training encoder-decoder models for bidirectional translation. Similarly, *Emojinize* (Klein et al., 2024) explored text-to-emoji generation using prompt engineering, reporting that emojis composed with context significantly improved human guessability. These works motivate our focus on adding age-awareness to emoji translation.

3 Methodology

3.1 Model Architecture

We selected BART-Large (Bidirectional and Auto-Regressive Transformers) as the backbone model for fine-tuning in this project (Lewis et al., 2020).

- **Bidirectional Encoder:** It reads the entire emoji sequence at once (left-to-right and right-to-left), allowing it to capture the full context before starting the translation.
- **Autoregressive Decoder:** It generates the English translation one word at a time, ensuring the output is fluent and grammatically correct.

This architecture is ideal for our task because translating emojis requires both a deep understanding of the input symbols (Encoder) and the ability to generate natural-sounding English sentences (Decoder).

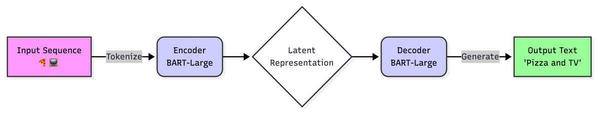


Figure 2: The BART-Large Encoder-Decoder architecture used for Emojisense

3.2 Tokenization and Vocabulary Expansion

A major challenge in this project is that standard models like BART are not optimized for emojis. Their pre-existing vocabularies view emojis as "unknown" tokens (<unk>), meaning the model cannot distinguish between a pizza emoji and a rocket emoji (Peng et al., 2023).

To fix this, we manually expanded the BART tokenizer by adding over 1,300 unique emoji symbols to its vocabulary. We resized the model's embedding layer to accommodate these new tokens, allowing the model to learn a specific vector representation for each emoji during training.

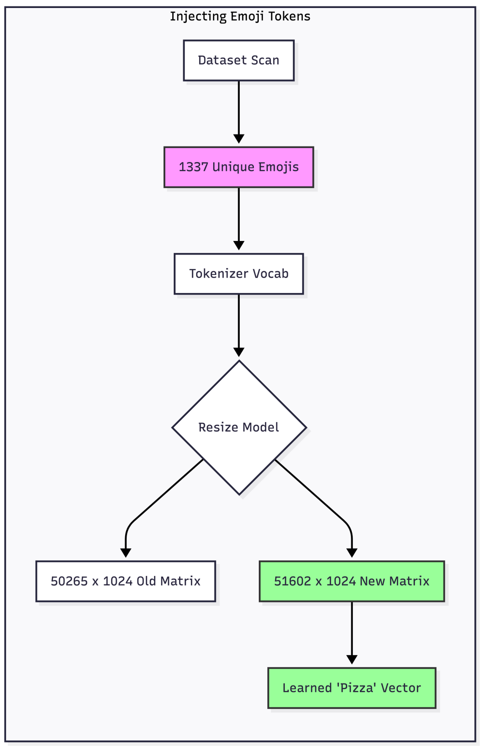


Figure 3: Expanding BART's emoji vocabulary

Handling Composed Emojis A distinct challenge in emoji translation is the concept of "composed emojis." These emojis are encoded as a sequence, with different symbols joined by a hidden character called the Zero Width Joiner (ZWJ or `\u200d`). For example, the 'Female Astronaut' emoji is technically a sequence of a 'Woman' emoji combined with a 'Rocket' emoji. To ensure our translator captures these nuances, we tokenize composed emojis into their constituent parts, separating them with the "`\u200d`" token. We **explicitly** treat the ZWJ separator as a token to be learned as well, which helps the model grasp the intricate compositional relationships between symbols.

3.3 Persona-Aware Style Transfer

To enable the model (Step 3) to generate age-specific interpretations, we introduced **Control Tokens**. These are special persona markers added to the beginning of the input sequence that tell the model which style to generate.

- `<STD>`: Standard English translation.
- `<GEN_Z>`: Casual, slang-heavy translation.
- `<BOOMER>`: Formal, enthusiastic translation.

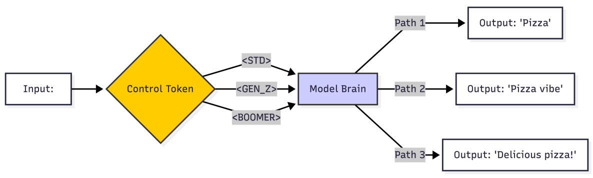


Figure 4: The control tokens

Smart Initialization When these control tokens were first introduced, their embeddings were randomly initialized. This led to training instability and eventual model collapse, in which the model became confused and began producing repetitive, low-quality (degenerate) outputs. To prevent this, we used a technique called **Smart Initialization**. Instead of starting from scratch, we manually copy the embedding weights from semantically similar English words to initialize the new control tokens. For instance, we initialized the embedding of the `<GEN_Z>` token using the pre-existing weights for the word "teenager," and the `<BOOMER>` token using the weights for "formal." This gave the model a hint to start with, allowing it to converge much faster during training.

3.4 Evaluation Metrics

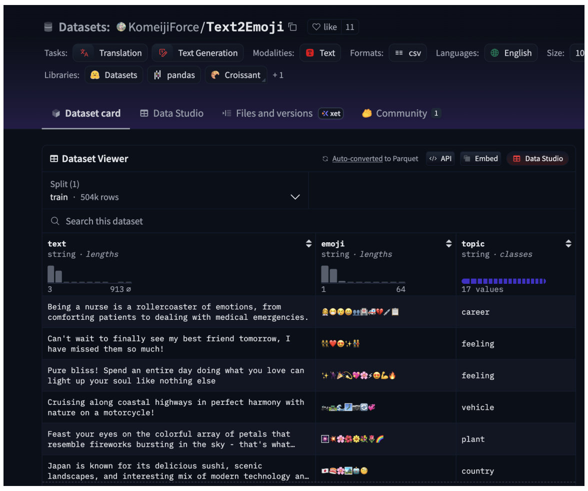
To measure our success, we use two complementary metrics:

- **BLEU Score:** Measures the precision of the translation by checking for exact word overlap with the reference text (Wikipedia).
- **BERTScore:** Measures semantic similarity. Since "I'm dead" and "I'm dying of laughter" mean the same thing but share few words, BLEU might score them low. BERTScore uses embeddings to verify that the *meaning* is preserved, which is crucial for evaluating style transfer (Zhang et al., 2019).

4 Dataset

4.1 Data Filtering and Preprocessing (Step 1)

We derived our primary dataset from the open-source `text2emoji` corpus available on Hugging Face (Peng et al., 2023). This dataset contains approximately 504,000 examples, where each entry consists of a text sequence, its corresponding emoji translation, and a topic category (Figure 5).


Datasets: KomeijiForce/Text2Emoji like 11
Tasks: Translation Text Generation Modalities: Text Formats: csv Languages: English Size: 10
Libraries: Datasets pandas Croissant +1
Dataset card Data Studio Files and versions xet Community 1
Dataset Viewer Auto-converted to Parquet API Embed Data Studio
Split (1)
train · 504k rows
Search this dataset

text string · *lengths* 3 913 ⌀	emoji string · *lengths* 1 64	topic string · *classes* 17 values
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
Feast your eyes on the colorful array of petals that resemble fireworks bursting in the sky - that's what…	🎆💥🌸🌺🌼💐🌷🌈	plant
Japan is known for its delicious sushi, scenic landscapes, and interesting mix of modern technology an…	🇯🇵🍣🌸🏞️🤖🧐	country

Figure 5: The text2emoji dataset

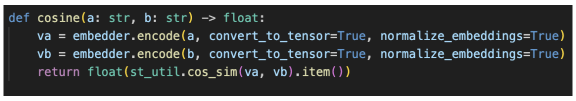
A significant limitation of this corpus is its synthetic nature; it was generated entirely by ChatGPT-3.5, meaning that the ground-truth labels reflect the interpretations of a single model (Peng et al., 2023) and may therefore contain inaccuracies or hallucinations. To mitigate this issue, we performed **cross-validation** on the dataset using a secondary LLM. Specifically, we selected **Mistral-7B**, an open-source model (Mistral, 2025), to act as our verifier.

Our validation process proceeds as follows:

1. We feed the emoji sequences into Mistral via API and prompt it to reverse-translate them back into English.

2. We encode both the original ground-truth text and Mistral's reverse translation into vector embeddings using the `sentence_transformers` library.

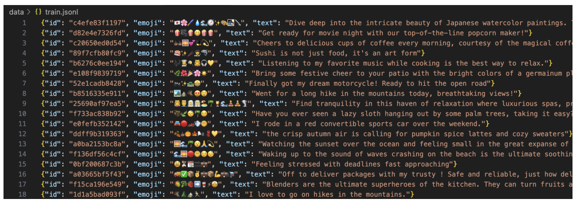
3. We compute the cosine similarity between these two embedding vectors:



```
def cosine(a: str, b: str) -> float:
    va = embedder.encode(a, convert_to_tensor=True, normalize_embeddings=True)
    vb = embedder.encode(b, convert_to_tensor=True, normalize_embeddings=True)
    return float(st_util.cos_sim(va, vb).item())
```

Figure 6: Cosine similarity formula used for validation

While a cosine similarity above 0.5 typically indicates semantic relatedness (Krantz and Jonker, 2024), we selected a stricter threshold of 0.65 to filter out low-quality or ambiguous samples. Data samples falling below this threshold were discarded. The remaining validated samples were stored in JSONL format (Figure 7), containing a unique ID, the emoji sequence, and the text reference.



```
data > {} train.jsonl
{"id": "c4efe83f1197", "emoji": "🇯🇵🌸🖌️💧🌊🧭✨🎨🖼️🖌️", "text": "Dive deep into the intricate beauty of Japanese watercolor paintings. 
{"id": "d82e4e7326fd", "emoji": "🍿🎥🍿😋🍿🍿", "text": "Get ready for movie night with our top-of-the-line popcorn maker!"}
{"id": "c20650ed0d54", "emoji": "☕☕🌅💕🎩💫", "text": "Cheers to delicious cups of coffee every morning, courtesy of the magical coff
{"id": "89f7cfb80fc9", "emoji": "🍣✨🥢👨‍🎨🐬", "text": "Sushi is not just food, it's an art form"}
{"id": "b6276c0ee194", "emoji": "🎶👨‍🍳🍳🙆🎧💛", "text": "Listening to my favorite music while cooking is the best way to relax."}
{"id": "e108f9839719", "emoji": "🌿🌺🎉🌸☀️", "text": "Bring some festive cheer to your patio with the bright colors of a germainum pl
{"id": "52e1cadb8428", "emoji": "🏍️✨🛣️🤩", "text": "Finally got my dream motorcycle! Ready to hit the open road"}
{"id": "b8516335e911", "emoji": "🏞️⛰️🥾😍😲", "text": "Went for a long hike in the mountains today, breathtaking views!"}
{"id": "25690af97ea5", "emoji": "💆💆‍♂️🧖‍♀️🧖‍♂️🏝️🌴🍷🌊🧘🧘🌪️", "text": "Find tranquility in this haven of relaxation where luxurious spas, pr
{"id": "f733ac838b92", "emoji": "🦥🌿😴🌴😌", "text": "Have you ever seen a lazy sloth hanging out by some palm trees, taking it easy
{"id": "e0fefb352142", "emoji": "🚘🔴🏎️💨😊", "text": "I rode in a red convertible sports car over the weekend."}
{"id": "ddff9b319363", "emoji": "🍂☕🎃🍁🌬️🧣💛", "text": "the crisp autumn air is calling for pumpkin spice lattes and cozy sweaters"}
{"id": "a0ba2153bc8a", "emoji": "🌅🌊🌴😮🙏💫", "text": "Watching the sunset over the ocean and feeling small in the great expanse of 
{"id": "f136df56c4cf", "emoji": "🌊🌅🔴😊😌😴", "text": "Waking up to the sound of waves crashing on the beach is the ultimate soothin
{"id": "0bf200687c3b", "emoji": "😩⏳📅➡️💼", "text": "Feeling stressed with deadlines fast approaching"}
{"id": "a03665bf5f43", "emoji": "🚚✅📦✌️💼📦💪💼📭", "text": "Off to deliver packages with my trusty ! Safe and reliable, just how del
{"id": "f15ca196e549", "emoji": "🥭🥦🍓➡️🥤⚡🤩", "text": "Blenders are the ultimate superheroes of the kitchen. They can turn fruits a
{"id": "1d1a5bad093f", "emoji": "🥾🌲⛰️🏃", "text": "I love to go on hikes in the mountains."}
```

Figure 7: Structure of the final filtered dataset

Given the computational intensity of encoding and verifying half a million sequences, we utilized the USC Center for Advanced Research Computing (CARC) to execute the cross-validation process (USC). This yielded a curated, high-quality dataset of 100,000 filtered samples. We partitioned this final dataset into an 80/10/10 split for training, validation, and testing, respectively, which served as the foundation for our baseline model in step 1.

4.2 Context and Modality Augmentation (Step 2)

Although the filtered dataset from Step 1 provided a strong vocabulary foundation, it remained topic-siloed. Due to the original design of `text2emoji`, which partitioned examples by topic, instances never combined concepts from different domains. As a result, the model's ability to handle more complex, multi-concept narratives was limited, making it primarily effective only for single-emoji prediction.

To address this limitation, we designed a **synthetic augmentation** process with two primary objectives: (1) combining examples from different topical domains, and (2) enabling mixed-modal understanding of text–emoji inputs.

- **Domain Mash-up Strategy:** We developed a script to randomly sample two unrelated examples from the dataset (e.g., a "Food" entry and an "Activity" entry) and concatenate them using natural conjunctions. This produces more complex training instances, encouraging the model to attend to the entire mixed-context emoji sequence rather than overfitting to a single dominant symbol.

- **Mixed-Modal Injection:** Concurrently, we introduced mixed text–emoji inputs. By reverse-engineering the dataset, we randomly injected emoji tokens into standard English sentences. This encourages the model to jointly interpret text and emojis as a unified semantic representation, while allowing surrounding text to provide contextual cues for emoji disambiguation.

Using this augmentation process, we constructed a synthetic dataset of 100,000 examples: 15,000 generated via the domain mash-up strategy, 15,000 via mixed-modal injection, and 70,000 randomly sampled from the Step 1 dataset to preserve learning of pure emoji semantics. The dataset was split into training, development, and test sets using an 8/1/1 ratio and was used to fine-tune the model in Step 2.

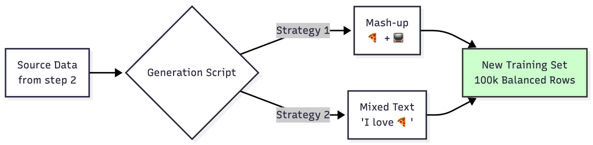


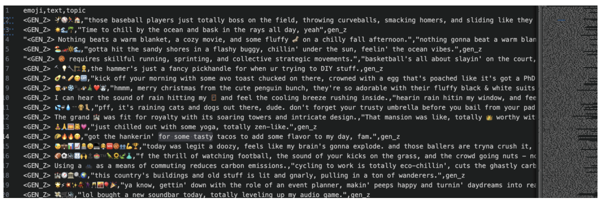
Figure 8: The augmentation process visualized

4.3 Stylistic Data Generation (Step 3)

The main challenge for Step 3 was that there is no existing dataset for "Gen Z" or "Boomer" emoji translations. To solve this, we created our own data using a **Teacher-Student** approach.

- **The Teacher:** We utilized **Microsoft Phi-3** (Microsoft, 2024), a high-performance Large Language Model, to act as a "cultural translator."

- **The Process:** We prompted Phi-3 to rewrite sentences from our base dataset (from step 2) into specific personas using instructions such as: *"Rewrite this sentence as a Gen Z teenager using internet slang and irony"* or *"Rewrite this as an enthusiastic Baby Boomer using formal language."*
- **The Student Data:** We collected these outputs to build a style-specific dataset of 40,000 examples (20,000 Gen Z and 20,000 Boomer). We combined these with 60,000 standard examples to create the final dataset (split into train/dev/test by 8/1/1 ratio) used to fine-tune the model in Step 3. This mix ensures the model retains the ability to translate normal English while our synthetic data teaches the Student model (BART-Large) to master different tones and styles across different age personas.



```
emoji,text,topic
<GEN_Z> 🤸⚾🏃🏠,"those baseball players just totally boss on the field, throwing curveballs, smacking homers, and sliding like they
<GEN_Z> ☀️🌊🌴,"Time to chill by the ocean and bask in the rays all day, yeah",gen_z
"<GEN_Z> Nothing beats a warm blanket, a cozy movie, and some fluffy 🧦 on a chilly fall afternoon.","nothing gonna beat a warm blan
<GEN_Z> 🏝️🏎️☀️🌊,"gotta hit the sandy shores in a flashy buggy, chillin' under the sun, feelin' the ocean vibes.",gen_z
"<GEN_Z> 🏀 requires skillful running, sprinting, and collective strategic movements.","basketball's all about slayin' on the court,
<GEN_Z> ⛏️💡🔧🏗️👷,the hammer's just a fancy pickhandle for when ur trying to DIY stuff.,gen_z
<GEN_Z> 🥑🍳🥖😋🆙,"kick off your morning with some avo toast chucked on there, crowned with a egg that's poached like it's got a PhD
<GEN_Z> 🤶🐧❄️🐾🐧🎄❤️☃️,"hmmm, merry christmas from the cute penguin bunch, they're so adorable with their fluffy black & white suits
<GEN_Z> I can hear the sound of rain hitting my 🪟 and feel the cooling breeze rushing inside.,"hearin rain hitin my window, and fee
<GEN_Z> 💦☔💧☁️🧥👢,"pff, it's raining cats and dogs out there, dude. don't forget your trusty umbrella before you bail from your pad
<GEN_Z> The grand 🏰 was fit for royalty with its soaring towers and intricate design.,"That mansion was like, totally 👑 worthy wit
<GEN_Z> 🧘🙏🌅🐱💗,"just chilled out with some yoga, totally zen-like.",gen_z
<GEN_Z> 🌮🔥🔥😋,"got the hankerin' for some tasty tacos to add some flavor to my day, fam.",gen_z
<GEN_Z> 😓💼📊📈👨‍💻😩💻👩‍💼🆘🏀👥💪🏆,"today was legit a doozy, feels like my brain's gonna explode. and those ballers are tryna crush it,
<GEN_Z> 🏈⚽👟👬🙌⬆️🏟️™️🦜🌻🌿👗,"f the thrill of watching football, the sound of your kicks on the grass, and the crowd going nuts – no
<GEN_Z> Using a 🚲 as a means of commuting reduces carbon emissions.,"cycling to work is totally eco-chillin', cuts the ghastly carb
<GEN_Z> 🏰🧭🏛️🔍🌍,"this country's buildings and old stuff is lit and gnarly, pulling in a ton of wanderers.",gen_z
<GEN_Z> 🌟⚡💥✨💃🕺🥂🌆🎈🎉,"ya know, gettin' down with the role of an event planner, makin' peeps happy and turnin' daydreams into rea
<GEN_Z> 🎛️🛒🔊,"lol bought a new soundbar today, totally leveling up my audio game.",gen_z
```

Figure 9: Portion of the stylistic dataset we generated

5 Experiments

5.1 Experimental Setup and Training

All three models were implemented using the `Hugging Face Transformers` library and trained on the USC Center for Advanced Research Computing (CARC) cluster (USC).

We fine-tuned three variants of the **BART-Large** model (Lewis et al., 2020), corresponding to the three training stages described in the Introduction and Dataset sections. Model 1 was trained on the filtered baseline dataset (Section 4.1), Model 2 on the augmented mixed-modal dataset (Section 4.2), and Model 3 on the stylistic, age-aware dataset (Section 4.3). To ensure a fair comparison, we used consistent hyperparameters across all three stages:

- **Optimizer:** AdamW with a weight decay of 0.01.
- **Learning Rate:** $2e^{-5}$ with a linear learning rate scheduler.
- **Batch Size:** 32
- **Epochs:** Each stage was trained for 5 epochs, stopping early if validation loss did not improve for 2 consecutive evaluations.
- **Gradient Clipping:** Set to 0.5 to prevent exploding gradients, which we found critical during the initial token embedding resizing phase.

For the **Step 3 (Age-Aware)** model, we froze the lower layers of the encoder during the first epoch to allow the new control token embeddings (`<STD>`, `<GEN_Z>`, `<BOOMER>`) to align with the pre-trained weights without destabilizing the model. Combined with smart initialization, this strategy helped prevent training instability and model collapse.

5.2 Evaluation Protocol

To ensure a comprehensive evaluation across all intended interpretation styles (Standard, Gen Z, and Boomer), we constructed a composite test set using unseen data from each of our three dataset categories. This approach allows us to measure performance not just on general translation, but specifically on the model's ability to handle the "Gen Z" and "Boomer" personas.

Our final test set consists of 1,000 randomly shuffled examples:

- **600 Standard examples:** Sampled from the Step 2 test set to evaluate overall model accuracy under standard inputs.
- **200 Gen Z examples:** Sampled from the synthetic Gen Z test set constructed in Step 3 to assess slang and informal language interpretation.
- **200 Boomer examples:** Sampled from the synthetic Boomer test set constructed in Step 3 to evaluate formal style adaptation.

We evaluated our models using two primary metrics mentioned in our methodology section:

1. **BLEU Score:** To measure the n-gram precision and fluency of the generated text.
2. **BERTScore (F1):** To measure semantic similarity. This is our primary metric for success, as it captures whether the model preserved the *meaning* of the emoji sequence. We report the F1 variant of BERTScore, which balances precision and recall.

Table 1: Performance comparison on the composite test set

Model Variation	BLEU Score	BERTScore (F1)
Model 1 (Baseline)	0.54	84.02
Model 2 (Mixed-Modal and Context-Aware)	6.33	87.11
Model 3 (Age-Aware)	**11.24**	**89.76**

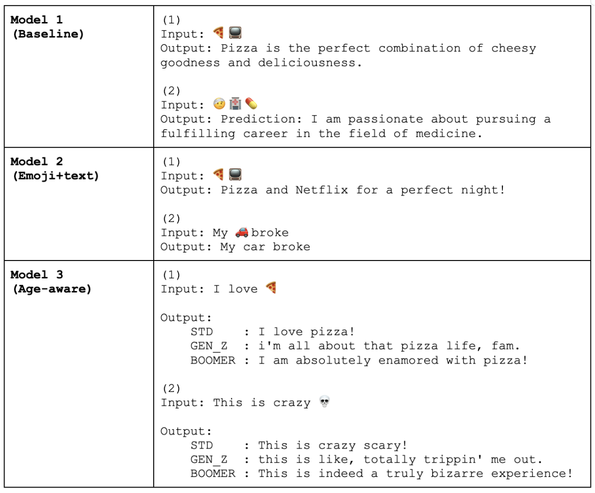
6 Results

6.1 Quantitative Results

Table 1 summarizes the performance of our three models on the composite test set. All metrics are reported on a normalized scale of 0–100 for clarity. We observe a consistent performance gain at each stage of the fine-tuning process. Model 1 (Baseline) yields the lowest scores across both metrics. Model 2 (Context-Aware) demonstrates a clear improvement, raising the BLEU score to 6.33 and BERTScore to 87.11. Finally, Model 3 (Age-Aware) achieves the best performance, reaching a BLEU score of 11.24 and a BERTScore of 89.76.

6.2 Qualitative Results

To visualize these improvements, Figure 10 presents selected examples of emoji-to-text translations generated by each of the three models for comparison. Additionally, we provide an interactive web demonstration of the final Age-Aware model (Model 3), which is accessible via the GitHub repository linked in the Appendix.



Model 1 (Baseline)	(1) Input: 🍕📺 Output: Pizza is the perfect combination of cheesy goodness and deliciousness. (2) Input: 🤕🏥💊 Output: Prediction: I am passionate about pursuing a fulfilling career in the field of medicine.
Model 2 (Emoji+text)	(1) Input: 🍕📺 Output: Pizza and Netflix for a perfect night! (2) Input: My 🚗broke Output: My car broke
Model 3 (Age-aware)	(1) Input: I love 🍕 Output: STD : I love pizza! GEN_Z : i'm all about that pizza life, fam. BOOMER : I am absolutely enamored with pizza! (2) Input: This is crazy 💀 Output: STD : This is crazy scary! GEN_Z : this is like, totally trippin' me out. BOOMER : This is indeed a truly bizarre experience!

Figure 10: Comparison of input-output pairs for the three models

7 Analysis and Discussion

7.1 Key Findings

Our results confirm our primary hypothesis: incorporating context and age-awareness significantly improves emoji translation quality. As shown in Table 1, Model 3 (Age-Aware) outperforms the baseline across all metrics.

The "Open-Ended" Nature of Translation We observe that BLEU scores are relatively low across all three models (Model 3 maxing at 11.24). This is expected given the open-ended nature of the task. Unlike translating French to English, where grammar is rigid, interpreting emoji sequences is highly subjective. A `<Fire>` emoji could validly translate to "fire," "lit," "awesome," or "hot," yet BLEU penalizes any deviation from the single ground truth reference. Therefore, while the relative improvement of Model 3 over the baseline (+10.7 BLEU) is meaningful, absolute BLEU values should not be interpreted in isolation.

Semantic Consistency (BERTScore) For this reason, we draw attention to **BERTScore** as the primary indicator of success, as it measures semantic meaning rather than exact word overlap.

- Model 1 achieved a baseline of 84.02, suggesting it captured basic keywords but missed the narrative.
- Model 3 achieved a score of **89.76**. In the context of BERTScore, where differences of 1–2 points often signify major improvements in model quality, a gain of +5.74 points represents a substantial leap in semantic understanding.

This validates that our Age-Aware model is not just guessing words, but successfully capturing the intended meaning of the input, even when the phrasing differs from the reference.

7.2 Qualitative Analysis

The qualitative outputs (Figure 10) mirror the quantitative gains and clearly illustrate the progress we made through each stage of fine-tuning.

- **Baseline Limitations (Step 1):** Model 1 performs adequately on single-emoji inputs but fails to generalize to emoji sequences. The sample results indicate that the model overfits to the most dominant emojis in the input while ignoring others (e.g., translating only the pizza emoji while disregarding the television emoji). This behavior is consistent with the limitation discussed in Section 4.2, where the Step 1 dataset is topic-siloed and restricts the model's ability to translate multi-concept inputs. The near-zero BLEU score further reflects the model's inability to generate coherent sentences from mixed emoji sequences.
- **Context Awareness (Step 2):** Through the introduction of the domain mash-up strategy, Model 2 acquired the ability to translate disparate concepts within a single input sequence. It successfully translates full emoji sequences (e.g., "Pizza and Netflix") rather than producing disjointed outputs, demonstrating improved contextual understanding. In addition, Model 2 supports mixed text–emoji inputs, representing a key improvement over Step 1 and providing a necessary foundation for the age-aware modeling introduced in Step 3.
- **Persona Adaptation (Step 3):** Finally, Model 3 demonstrates the ability to adapt its outputs to different user personas. It retains the contextual understanding developed in Model 2 while additionally modulating linguistic style based on age. Specifically, the model dynamically adjusts punctuation, capitalization, and lexical choices, producing informal grammar, slang, and nonstandard capitalization for Gen Z outputs (e.g., "i'm," "fam"), and more formal grammar, spelling, and capitalization for Boomer outputs (e.g., "bizarre"). At the same time, it continues to generate standard, neutral translations when prompted accordingly. This behavior confirms that the model successfully learned persona-conditioned style switching, a capability further reflected by its achieving the highest BERTScore among all three models.

8 Conclusion

8.1 Limitations

While our results are promising, EMOJISENSE faces two primary limitations. First, despite manually expanding our vocabulary, we cannot guarantee coverage for every emoji in the Unicode standard. This leads to potential Out-of-Vocabulary (OOV) errors, where the model may produce unpredictable inferences when encountering rare or newly released emojis. Second, our "Teacher-Student" loop in step 3 relies entirely on the capabilities of the Teacher LLM (Microsoft Phi-3). If the teacher model hallucinates or exhibits bias during data generation, these flaws are distilled into our student model, effectively capping our model's accuracy at the teacher's ceiling.

Another limitation is the simplicity of our age-group labeling. In this work, we only distinguish between Gen Z and Boomer users. While this is sufficient to demonstrate age-aware behavior, it does not capture the full range of language variation across different generations. Using more detailed age groups, such as Millennials or Gen X, or even finer age ranges, could allow the model to adapt its style more precisely.

In addition, the quality of the synthetic data is limited by the language models used for data generation. For practical and cost reasons, we relied on open-source LLMs. Although effective, using stronger language models could produce higher-quality and more diverse training data, which may further improve model performance.

8.2 Future Work

To address these limitations, we suggest several directions for future work. One important direction is **scaling and generalization**. Future work could increase the size of the training dataset, especially by including more rare and complex emoji sequences. Training the model for more epochs may also help it converge more smoothly and improve stylistic consistency.

Finally, the Teacher–Student loop in Step 3 could be made more robust by adding a **secondary validation filter**. Similar to the cross-LLM validation used during data preprocessing, an additional verifier model could be used to check the teacher's outputs and remove low-quality or hallucinated examples before training. This would help ensure that the student model learns from cleaner data and produces more reliable results.

References

Lars Henning Klein, Roland Aydin, and Robert West. 2024. Emojinize: Enriching any text with emoji translations. ArXiv:2403.03857.

Tom Krantz and Alexandra Jonker. 2024. What is cosine similarity? https://www.ibm.com/think/topics/cosine-similarity. Accessed: 2025-10-28.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Association for Computational Linguistics.

Microsoft. 2024. Phi-3 mini 128k instruct. https://huggingface.co/microsoft/Phi-3-mini-128k-instruct. Accessed: 2025-10-28.

Mistral. 2025. Models — getting started (documentation). https://docs.mistral.ai/getting-started/models. Accessed: 2025-10-28.

Letian Peng, Zilong Wang, Hang Liu, Zihan Wang, and Jingbo Shang. 2023. Emojilm: Modeling the new emoji language. ArXiv:2311.01751.

USC. Usc center for advanced research computing (carc). https://www.carc.usc.edu/. Accessed: 2025-10-28.

Philipp Wicke and João Miguel Cunha. 2020. An approach for text-to-emoji translation. In *Proceedings of the International Conference on Computational Creativity (ICCC'20)*, pages 167–171, Coimbra, Portugal. Association for Computational Creativity.

Wikipedia. Bleu – bilingual evaluation understudy (wikipedia). https://en.wikipedia.org/wiki/BLEU. Accessed: 2025-10-28.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv*.

Appendix

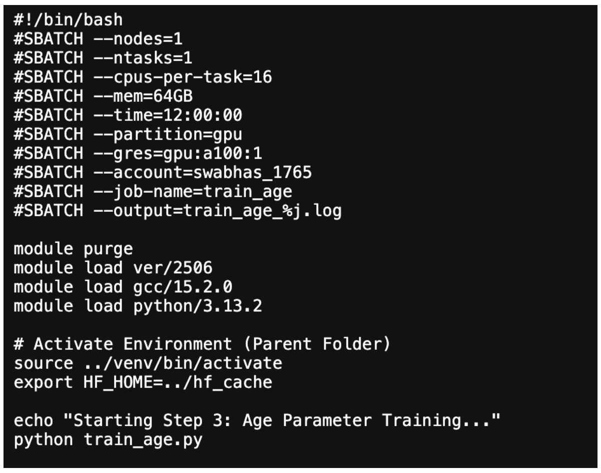
A Computational Infrastructure

A.1 CARC and Hardware Setup

All model training and fine-tuning were conducted using the Center for Advanced Research Computing (CARC). We specifically utilized the high-performance computing cluster equipped with **NVIDIA A100 GPUs**.

A.2 Job Scheduling

Resource allocation was managed using Slurm workload manager scripts. We configured `.slurm` batch scripts to define the partition, request specific GPU resources, allocate system memory, and set execution time limits. This ensured consistent environments for training runs and efficient use of the cluster's queue system.



```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --mem=64GB
#SBATCH --time=12:00:00
#SBATCH --partition=gpu
#SBATCH --gres=gpu:a100:1
#SBATCH --account=swabhas_1765
#SBATCH --job-name=train_age
#SBATCH --output=train_age_%j.log

module purge
module load ver/2506
module load gcc/15.2.0
module load python/3.13.2

# Activate Environment (Parent Folder)
source ../venv/bin/activate
export HF_HOME=../hf_cache

echo "Starting Step 3: Age Parameter Training..."
python train_age.py
```

Figure 11: sample slurm file used to submit jobs

B Project Resources & Reproducibility

To facilitate reproduction of our results, all code, data, and model artifacts have been made publicly available.

B.1 GitHub Repository

The complete source code is hosted on GitHub (https://github.com/KaitoPotato/EmojiSense). The repository is structured as follows:

- **Data Validation and Generation:** The exact prompts used to cross-validate and synthesize our training datasets are located in the `emojisense_src` directory.
- **Datasets:** The raw and processed data used

for training and testing are available within the repository.

- **Training:** Scripts detailing our tokenization strategy and fine-tuning loops for Models 1, 2, and 3 can be found in `emojisense_src`.
- **Interactive Demo:** The source code for the web-based demo (Model 3) is available in the `emojisense_demo` folder.