Overview

BuzzFeed quizzes are deterministic, often predictable, and fail to capture user nuance. The project started with the question, “Can free-form text be used with NLP techniques to infer Pokémon matches?” Rather than matching outcomes to answers, this project explores interpretations of meaning directly from natural language. The goal is to produce personalized results for the user, instead of pre-scripted.

Problem

1. The real problem is closing the gap between unstructured, variable inputs and semi-structured descriptive data in a way that preserves meaning. This project explores different NLP techniques to address that challenge, from word frequency methods, to vectorized semantic embeddings and transformers.
2. Text is unstructured and it doesn’t sit neatly in a database. It is more difficult to analyze variable human inputs than when answers are pre-matched.
3. Traditional NLP techniques do not capture semantic meaning. They typically rely on matching words and their frequency. More recent models can capture context, but are often computationally expensive, and take too long to give results.

Approach

1. Input
    a. Users enter free-form text based on the prompts
2. Processing
    a. The system encodes user data with both traditional and newer NLP techniques
    b. The system uses cosine similarity to compare the inputs to a preprocessed Pokédex corpus, ~9000 entries, aggregated ~1000 entries, prioritizing higher semantic relevance
3. Output
    a. The system outputs a team of Pokémon ordered by cosine similarity score

Hurdles

1. API
    a. Several endpoints
    b. Variable number of Pokédex entries per Pokémon
    c. Slow to call the API every time the system gets used
2. Model Performance vs User Experience
    a. BERT was prohibitively slow, no test with BERT successfully completed under acceptable runtime limits (often 20-30 minutes)
3. Results
    a. How could I tell which model was best without ground truth?

Overcoming

1. API
    a. Identifying necessary endpoints was a one time fixed cost task
    b. This required some custom aggregation logic and schema setup
    c. I preprocessed the Pokédex once, and compared those results to the smaller user input
2. Model Performance vs User Experience
    a. DistilBERT and MPNet are lightweight solutions, both process in seconds
3. Results
    a. I added a voting system to let the users tell me which model works best

Insights & Results

Method	Strengths	Weaknesses
TF-IDF	Fast, easy to explain the model, no training necessary	No semantic capture, ignores word order
BERT	Deep contextual understanding	Unusably slow, not easy to explain transformers
DistilBERT	Fast, good contextual understanding	Still fairly heavy, drop in performance from BERT
MPNet	Strong contextual understanding, mid computational cost	Higher computational cost than other models

TF-IDF and DistilBERT represent the lower and upper bounds of performance. TF-IDF ranges in the low 40% cosine similarity, and DistilBERT's cosine similarity ranges from low 80% to low 90% range, all while maintaining acceptable response times. MPNet trends just under DistilBERT's scores with similar processing times.

Next Steps

I have already received several pieces of feedback, such as including links to each Pokémon's Pokédex entry so the user can see the description. I tried a lightweight chatbot from HuggingFace to include a summary of what words and terms were good matches between input and comparison, but the results are not sufficiently reliable for deployment yet.

Takeaways

1. Data Engineering is tough work, even on good days.
2. Model quality does not equal product quality
3. Users prioritize fast, high quality results over model details, user experience constraints can outweigh marginal gains in model performance
4. Visualization tools are not always 1:1, especially when translating from one language to another

Why It Matters

This approach generalizes to recommendation systems, semantic search, and knowledge retrieval in production environments. This project shows end-to-end ML pipeline development, tradeoff analysis between model performance and model speed, and the challenges in moving from theoretical analytics to real production.