ANIMAL TALES AI

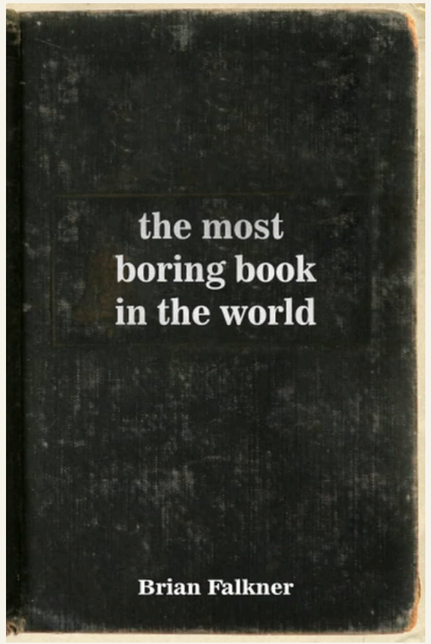
Presented By : Shaurya Bhardwaj

Thapar Institute of Engineering and Technology, Patiala, Punjab, India

Submitted to Dr. Raman Kumar Goyal

PROBLEM

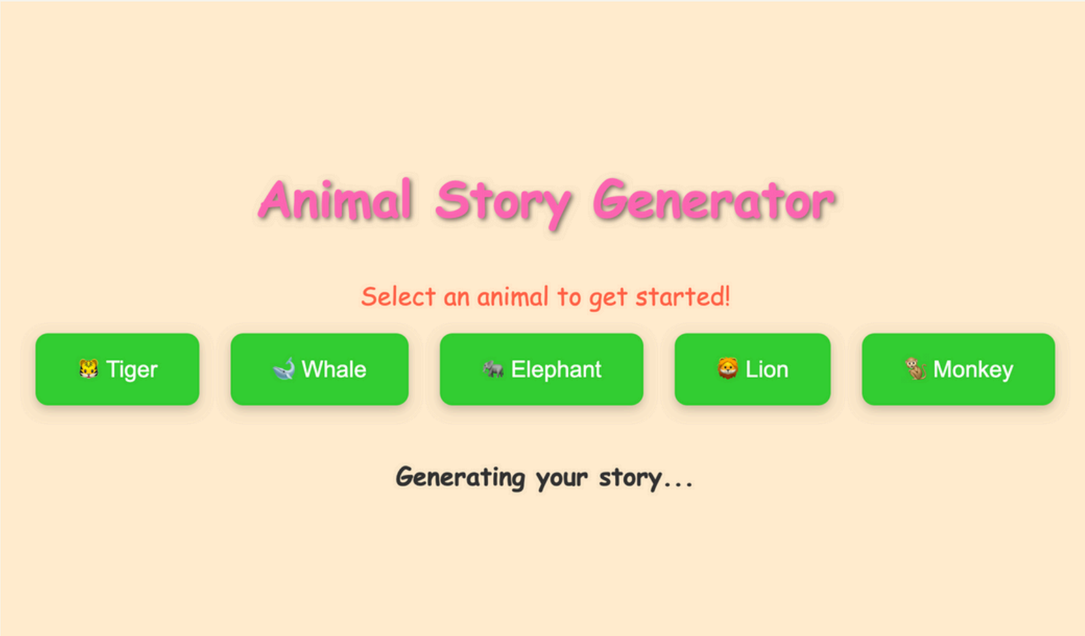
Most educational tools are either static (like textbooks or fixed storybooks) or provide limited stories. This makes it challenging to hold a child's attention and creativity over time.

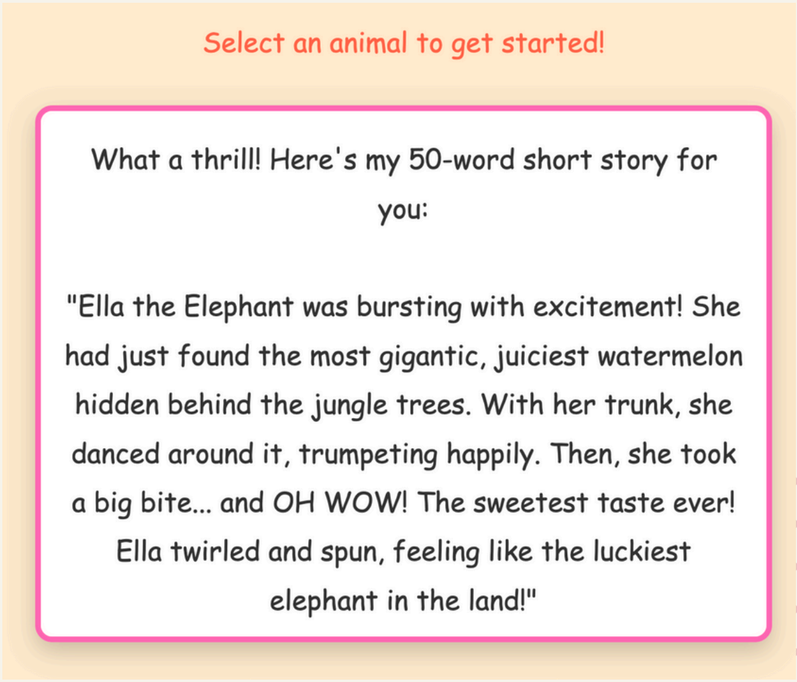


Children thrive when their inputs, preferences, and imagination are acknowledged. Dynamic tools that create tailored content can better nurture a child's creative thinking and curiosity.

PROPOSED SOLUTION

My tool takes simple inputs like animal names and moods and generates fully personalized stories that resonate with children's imagination.





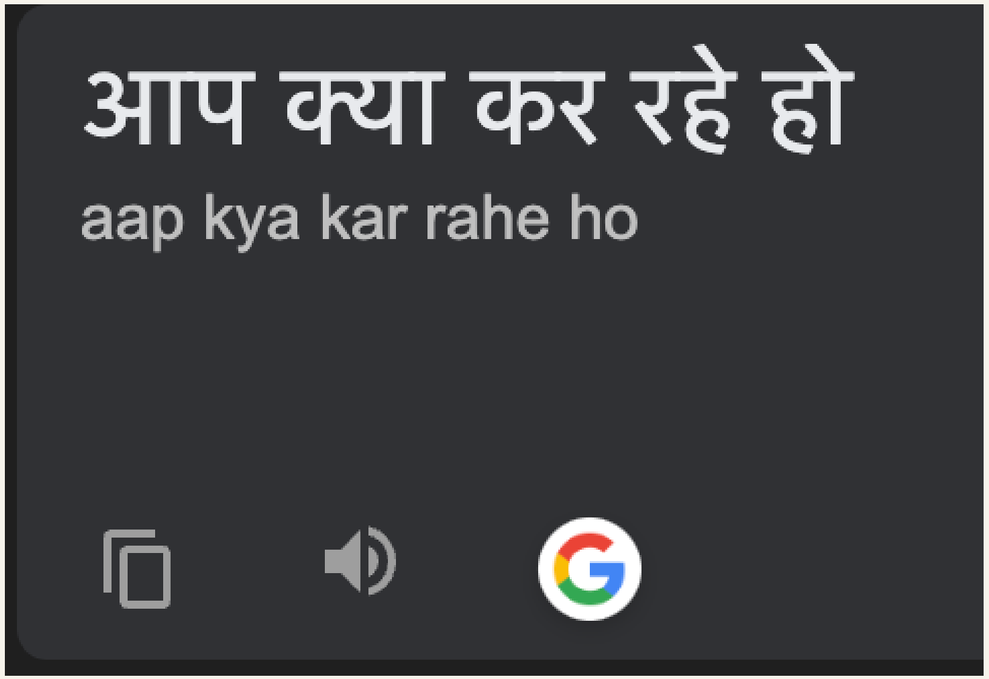
Allows children to actively participate in creating their stories, fostering a sense of involvement and ownership.

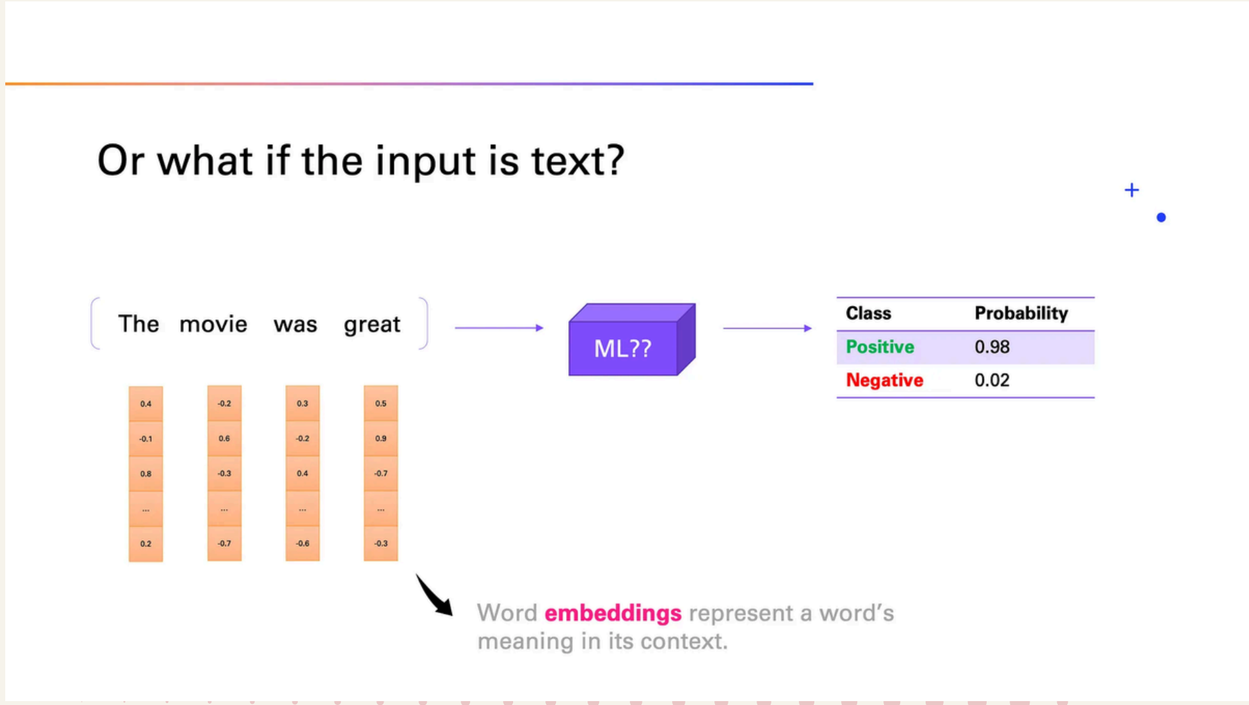
WHY ML SOLUTIONS ALONE CANNOT WORK

- Traditional machine learning models rely on information without context, which cannot work for creative tasks like storytelling.
- ML solutions lack linguistic understanding, one great example of this is translations from Hindi to English
- They can't understand abstract prompts like "make a funny story about a brave squirrel."
- While LLMs use something called Attention which is the core of the transformer architecture
- For these reasons, it doesn't make sense to compare llms with machine learning models.

WHY WE NEED LLMS

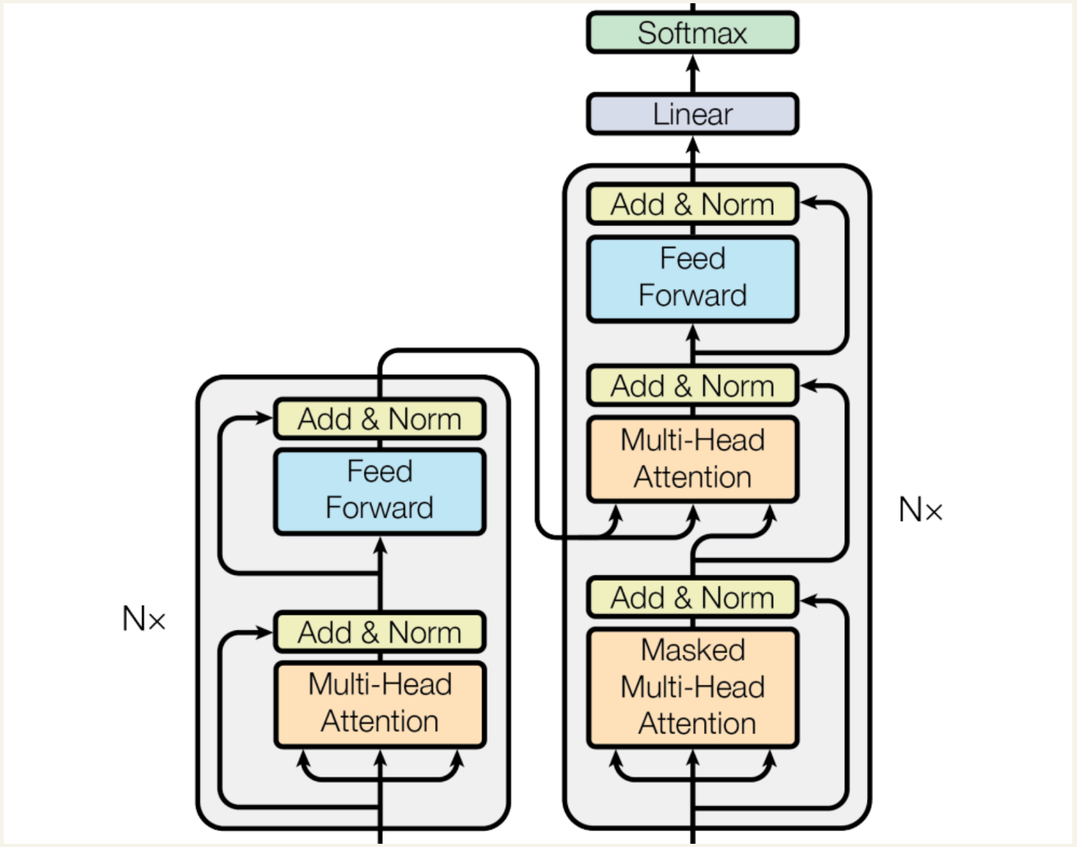
- In the domain of NLP, traditional text-processing techniques like Word Embeddings were used. Before models like BERT and Llama emerged. Transformers, starting with BERT, paved the way for LLMs, which are trained on vast amounts of text data from the internet. These models excel in complex linguistic tasks like translation, question-answering, and summarization, owing to their extensive training data and large parameter sizes.
- https://arxiv.org/pdf/1706.03762





HOW LLMS WORK

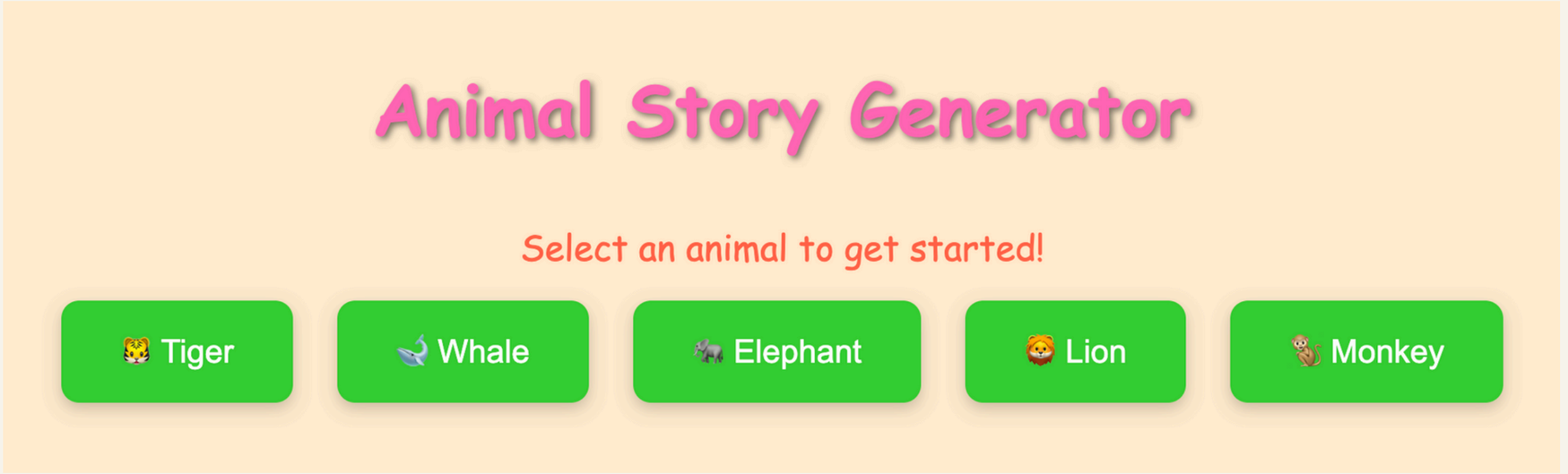
- LLMs are models trained on very very large data.
- For context, a 1 GB file can hold around 178 Million words and 1 petabyte is around 1 million GBs
- These are trained on huge architectures. Trained on Terabytes of data.
- Build on 2 things - Positional Encoding, Attention



$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

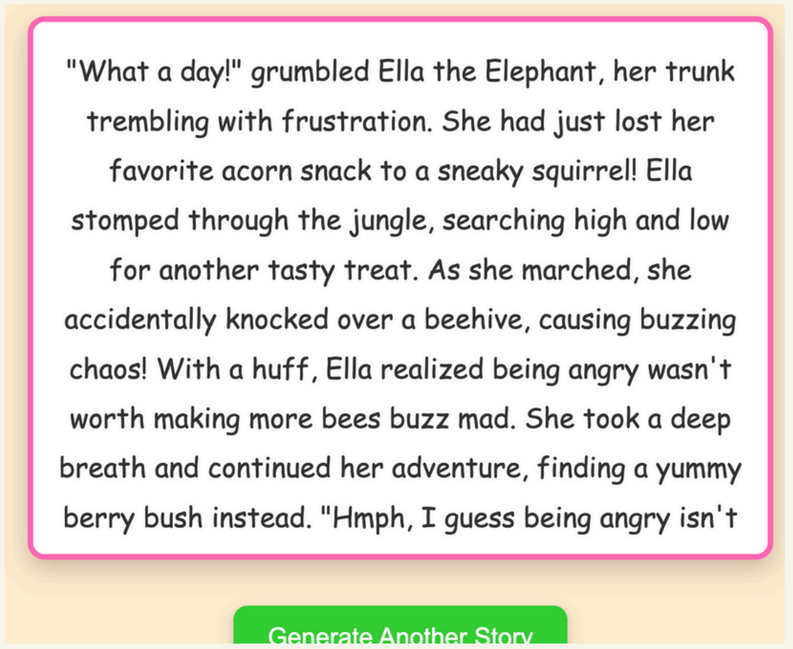
Model	Provider	Open-Source	Speed	Quality	Params	Fine-tuneability
gpt-4	OpenAI	No	★☆☆	★★★★	-	No
gpt-3.5-turbo	OpenAI	No	★★☆	★★★☆	175B	No
gpt-3	OpenAI	No	★☆☆	★★★☆	175B	No
ada, babbage, curie	OpenAI	No	★★★	★☆☆☆	350M - 7B	Yes
claude	Anthropic	Yes	★★☆	★★★☆	52B	no
claude-instant	Anthropic	Yes	★★★	★★☆☆	52B	No
command-xlarge	Cohere	No	★★☆	★★☆☆	50B	Yes
command-medium	Cohere	No	★★★	★☆☆☆	6B	Yes
BERT	Google	Yes	★★★	★☆☆☆	345M	Yes
T5	Google	Yes	★★☆	★☆☆☆	11B	Yes
PaLM	Google	Yes	★☆☆	★★☆☆	540B	Yes
LLaMA	Meta AI	Yes	★★☆	★★☆☆	65B	Yes
CTRL	Salesforce	Yes	★★★	★☆☆☆	1.6B	Yes
Dolly 2.0	Databricks	Yes	★★☆	★★☆☆	12B	Yes

VISUAL REPRESENTATION OF THE PROJECT




```
CMAKE_ARGS="-DLLAMA_CUBLAS=on" FORCE_CMAKE=1 pip install llama-cpp-python==0.1.78 numpy==1.23.4
pip install huggingface_hub
pip install llama-cpp-python==0.1.78
pip install numpy==1.23.4
to add a breakpoint
CostFunction = sphere  #
nVar =  of Decision Variables
VarSize  Size of Decision Variables Matrix
VarMin  r Bound of Variables
VarMax  Bound of Variables

w = 1 ia Weight
wdamp  # Inertia Weight Damping Ratio
c1 =  al Learning Coefficient
c2 =  l Learning Coefficient
```



SUMMARY AND FUTURE ENHANCEMENTS

- Developed a platform combining learning with fun storytelling.
- Empowered children to interact creatively with AI.

- Audio Narration: Add voice-based storytelling for inclusivity.
- Image as an input