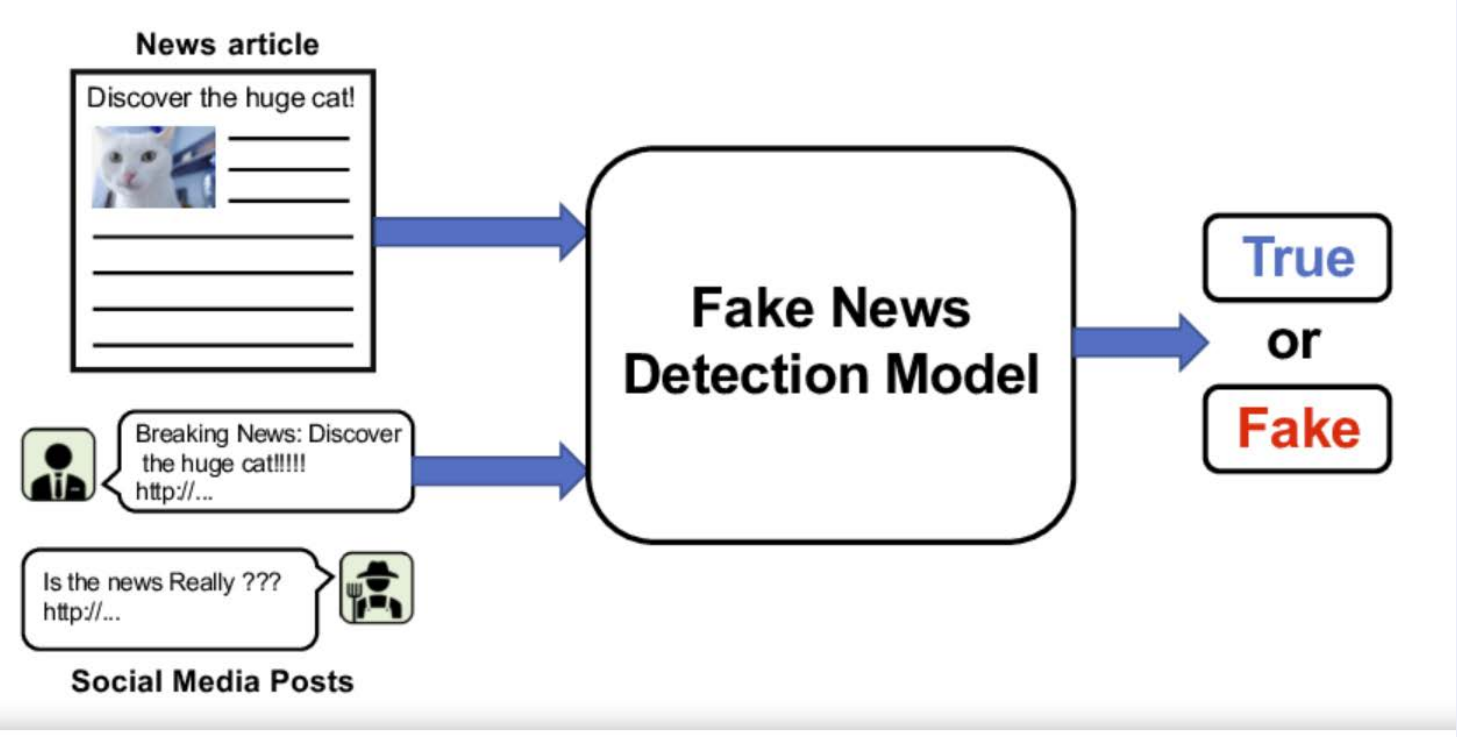
Fake News Detection by Supervised Fine Tuning of Tiny Llama2 Model

Wenhui Wang
Springboard

05/14/2024

The problem


News article
Discover the huge cat!
Breaking News: Discover the huge cat!!!!! http://...
Is the news Really ??? http://...
Social Media Posts
Fake News Detection Model
True
or
Fake

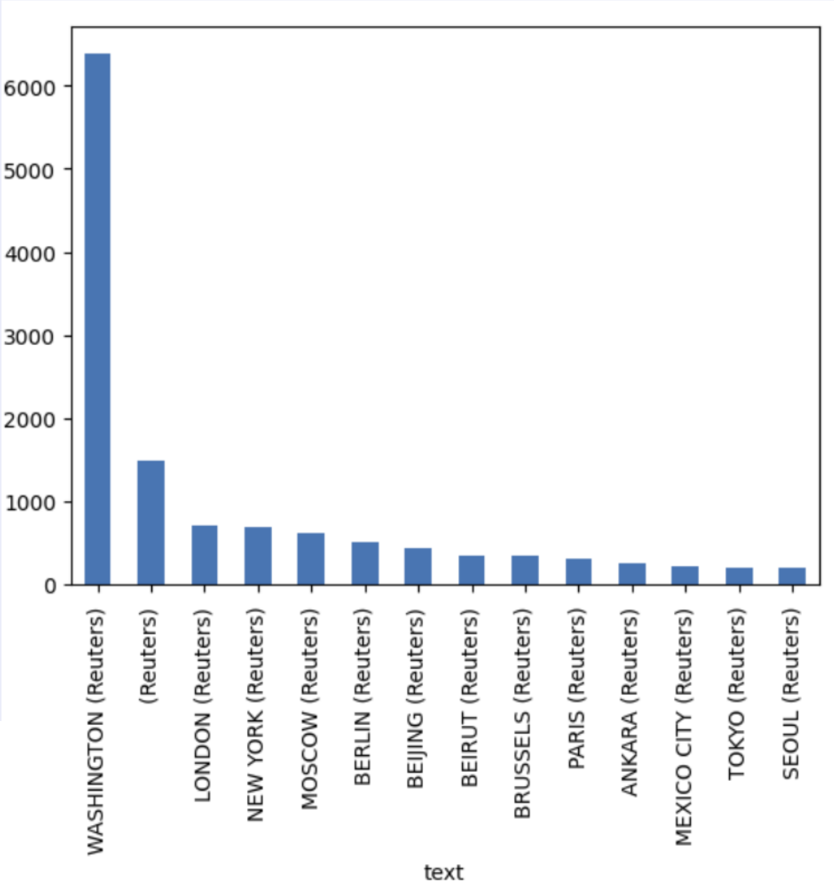
- Fake news specifically refers to news reports that are untrue or exaggerated. These reports may be deliberately created to mislead the public or promote a specific agenda, through traditional media channels like print, and television as well as non-traditional media channels like social media. Different traditional machine learning model has been built to detect fake news.

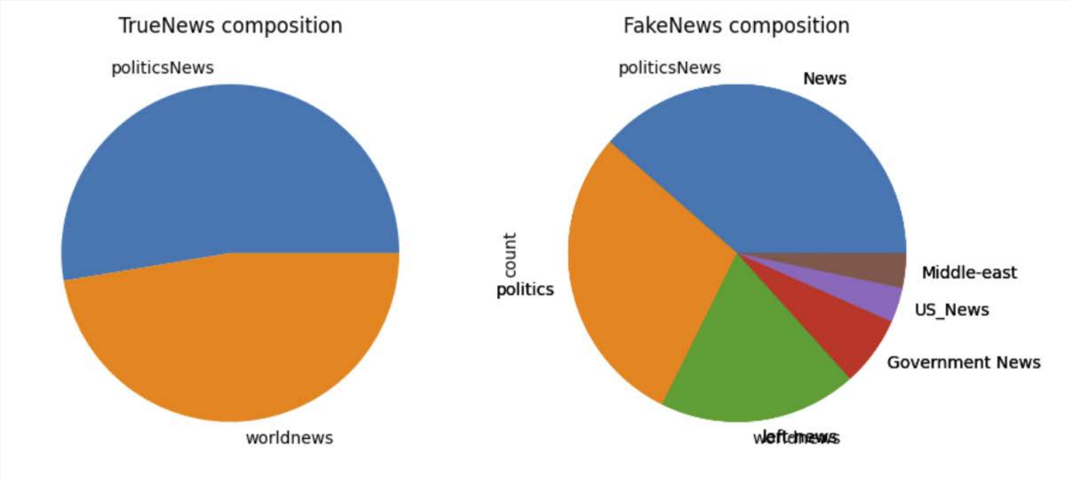
Targets of the project

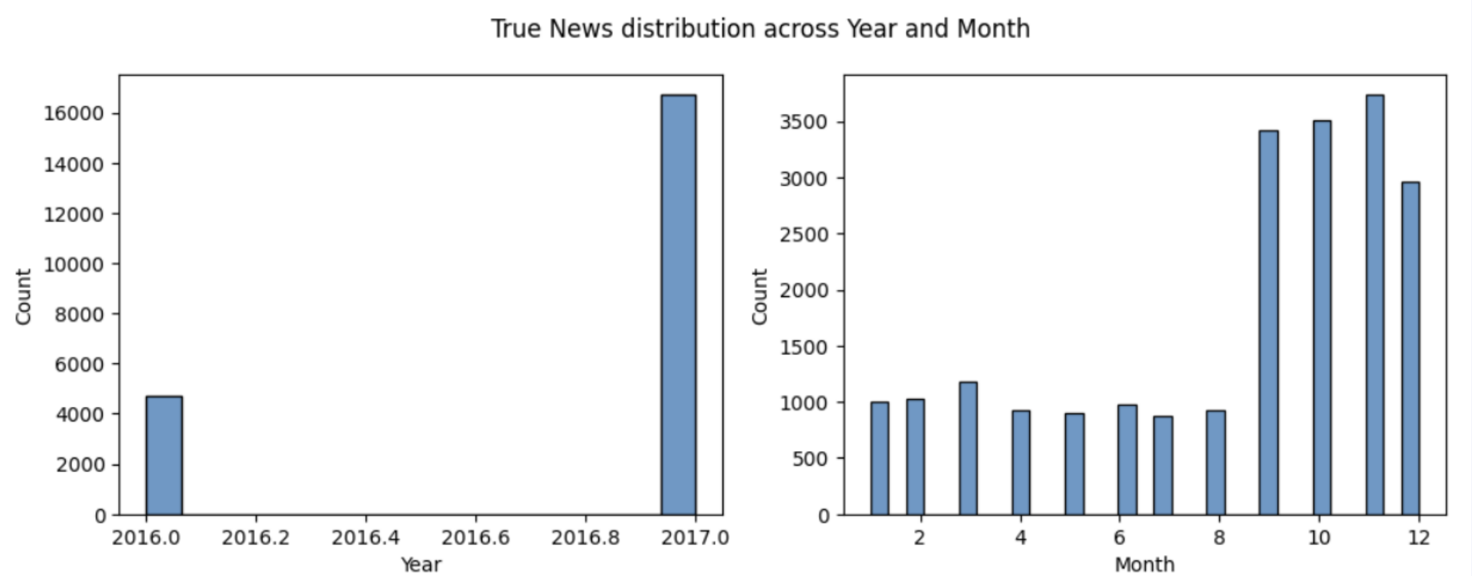
- Collect fake true news dataset and encode the text dtat with tf-idf. Made use tranditional machine learning to get the baseline performance.
- Fine tuning TinyLlama model to check large language model’s performance in fake news detection.

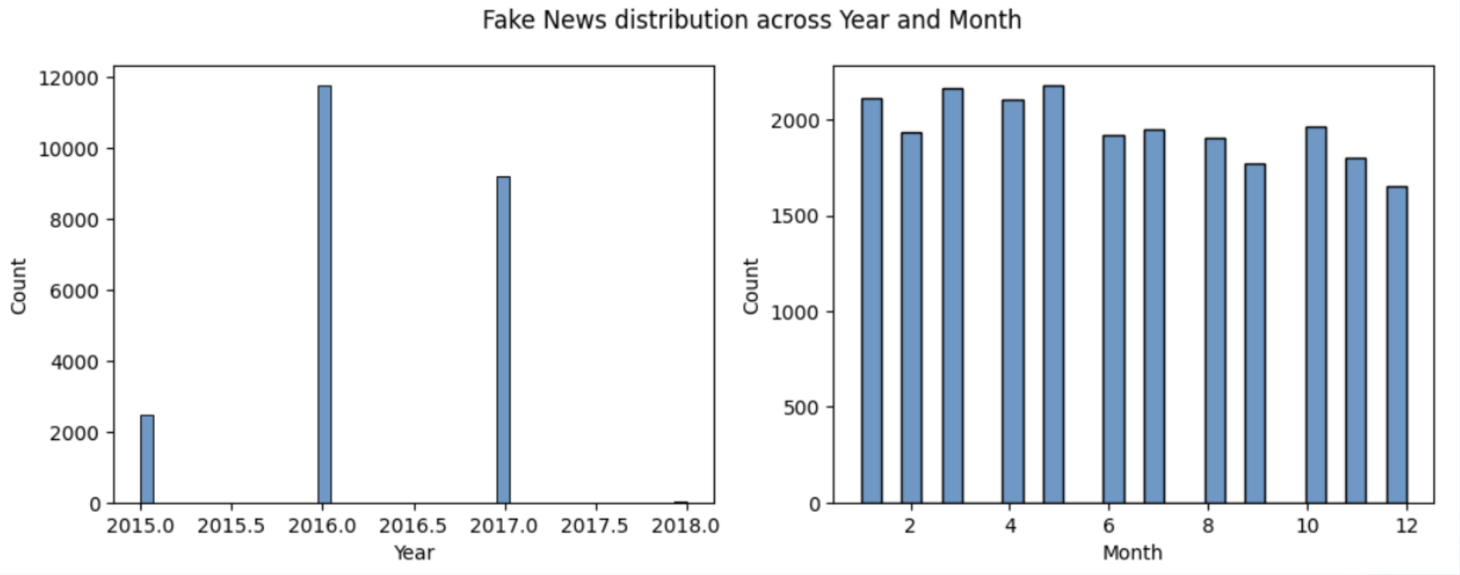
Data wrangling and clean

- Found two fake news detection dataset:
- The first dataset containting 21415 true news and 23481 fake news. All the true news are from Reuters. The exact site of Reuters is labeled in the ture news' text.
- Fake news are from different sources.



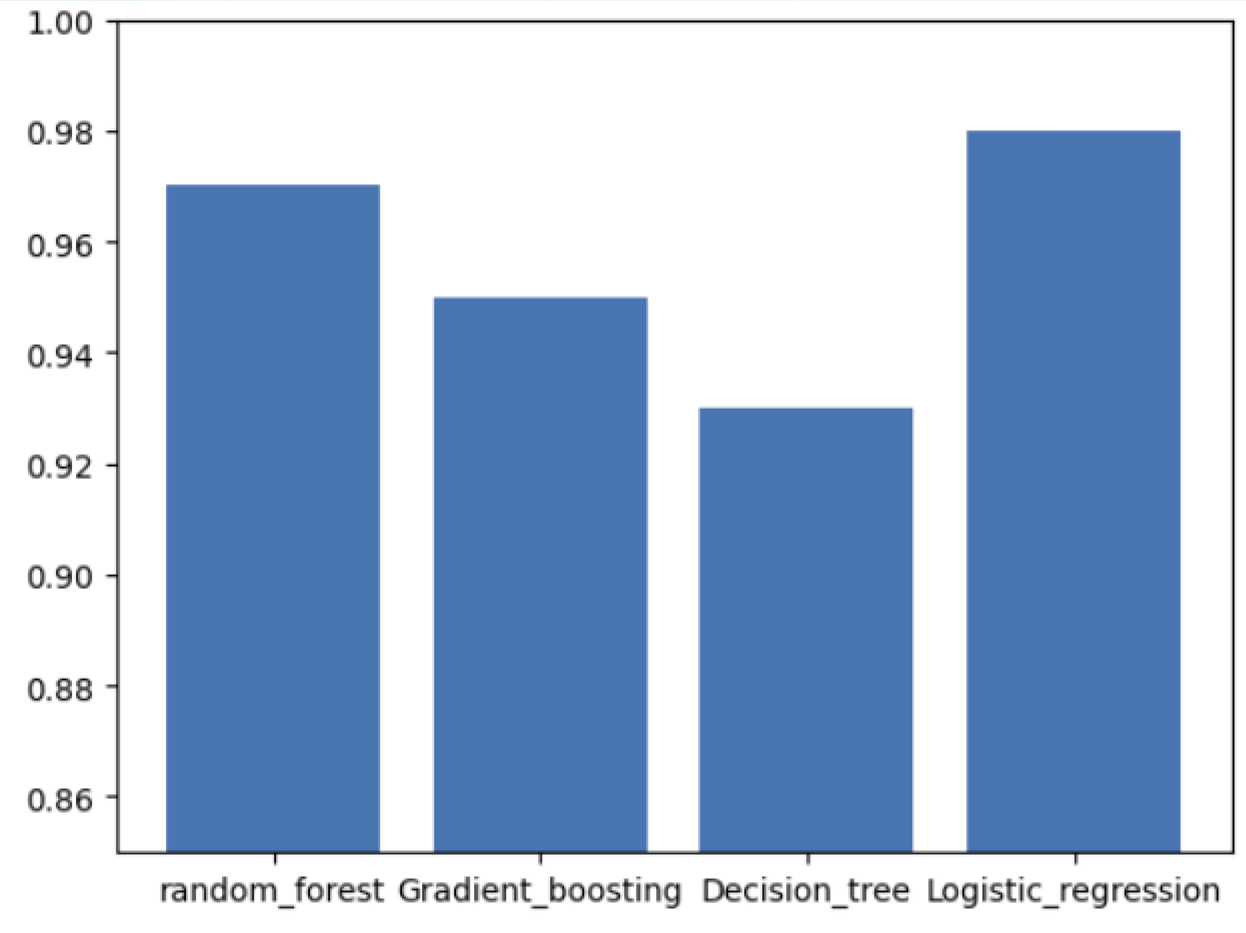






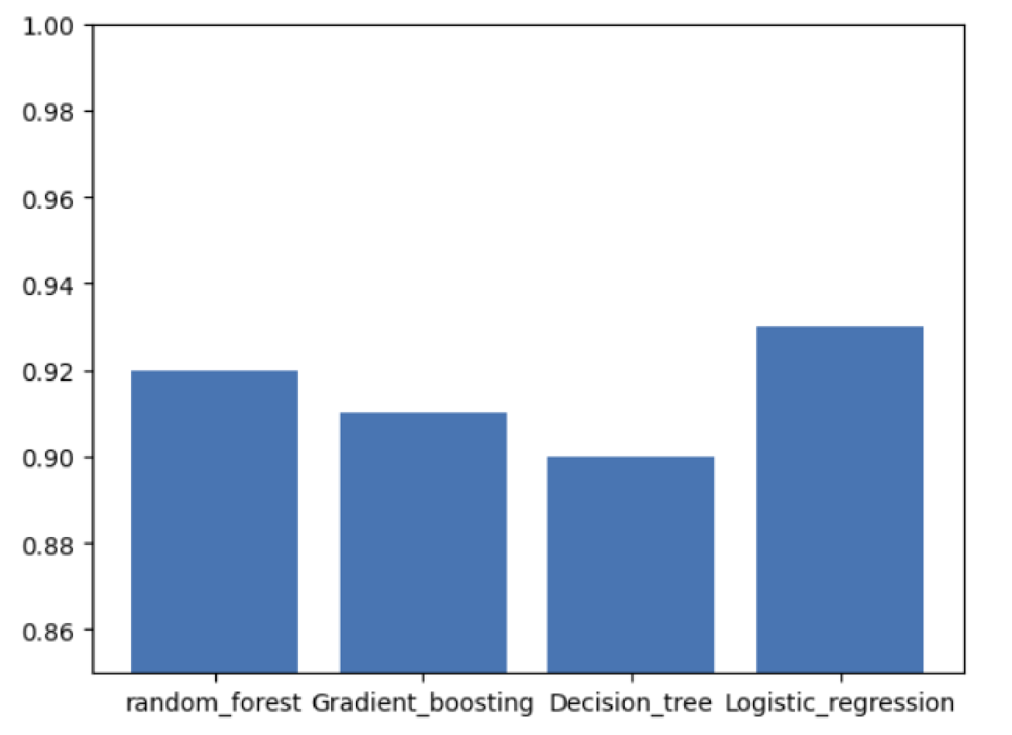
- True and fake news have different year and month distribuiton

- Performance with traditional ml model



- True news and fake news have significant meta data difference. Therefore the super performance can be impaired by other characteristics such as source, time and topic.
- This data is not appropriate for our project.

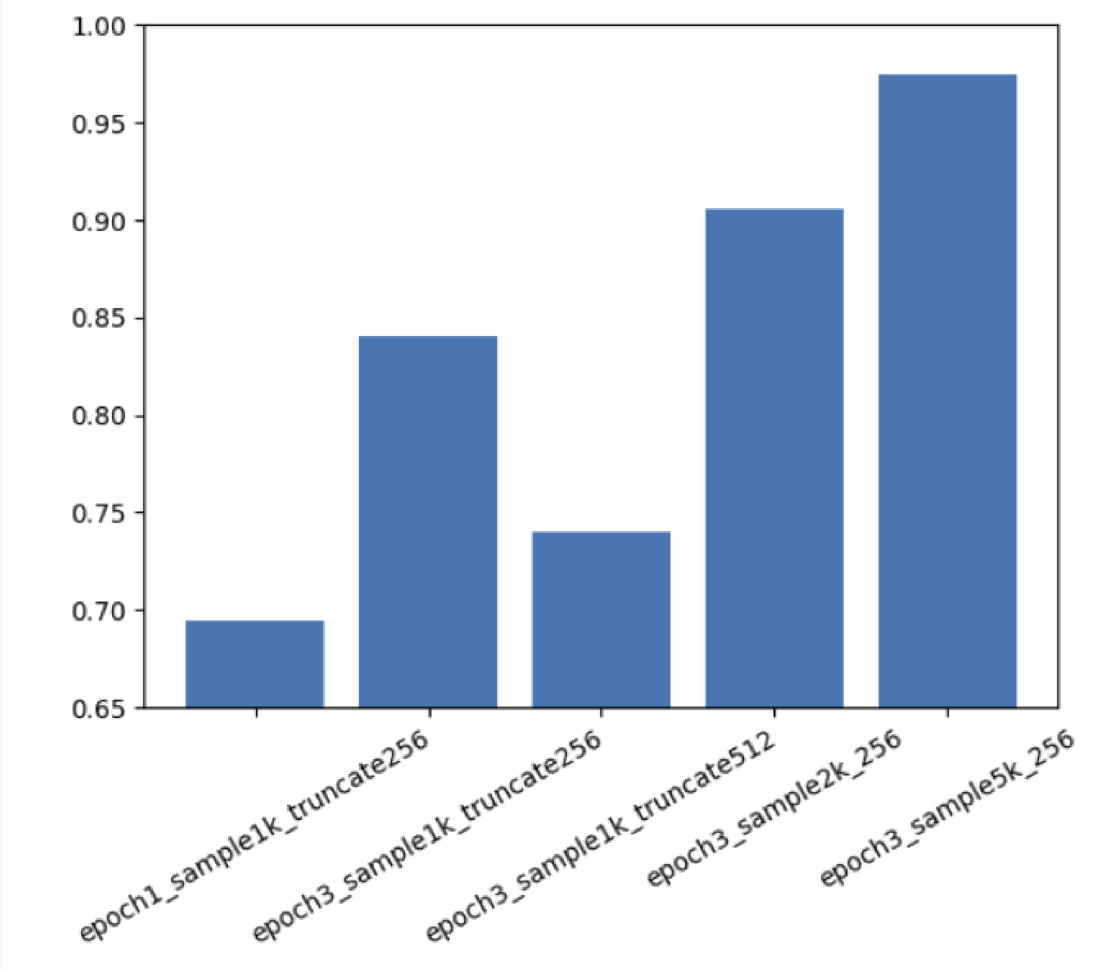
- I found another dataset. There are 34975 articles in true news file and 43642 in fake news file. There are no other meta data for the dataset.
- I run the same text preprocessing and encoding with tf-idf. After that, I checked the performance of traditional ML model on the data.



TinyLlama model

- TinyLlama is a compact 1.1B language model Building on the architecture and tokenizer of Llama 2.
- The Llama2 model was proposed in LLaMA: Open Foundation and Fine-Tuned Chat Models, which is a collection of pretrained and fine-tuned large language models (LLMs) ranging in scale from 7 billion to 70 billion parameters. Llama 2, like the original Llama model, is based on the Google transformer architecture, with improvements. Llama's improvements include RMSNorm pre-normalization, inspired by GPT-3; a SwiGLU activation function, inspired by Google's PaLM; multi-query attention instead of multi-head attention; and rotary positional embeddings (RoPE), inspired by GPT Neo. Llama training used the AdamW optimizer. Llama 2's primary differences from Llama are increased context length (4096 vs. 2048 tokens) and grouped-query attention (GQA) instead of multi-query attention (MQA) in the two larger models.
- Instead of focusing solely on training compute-optimal language models, inference-optimal language models, aiming for optimal performance within specific inference constraints, is achieved by training models with more tokens than what is recommended by the scaling law. Following the same architecture and tokenizer as Llama 2, TinyLlama is obtained by training transformer decoder-only model with 1.1B parameters using approximately 3 trillion tokens. TinyLlama model shows better performance comparing to large language with around 1B parameters. With its compact architecture and promising performance, TinyLlama can enable end-user applications on mobile devices.

- Following the idea of control computation cost and get optimal performance, I checked the contribution of number of trained samples, epochs, length of news.



- I choose to train the model with limited number of samples, which I tried 1000 and 2000. I also truncate the length of news to 256 and 512. I also tried to fine tuning the model with epoch = 1 and epoch =3
- From the figure, I found that:
- For the same number of samples and truncate size. Higher epochs give better performance.
- The plot also shows that longer truncate size gives worse performance. The reason is that the metric in training is to compare the generated text with true text. But the target of our problem is to classify the text to true and false. Longer text makes the classification section have even lower weight.
- We can also find that higher number of training samples present better performance.

Conclusion

- TinyLlama can present comparable or better performance with limited number of training data.
- Larger epochs present better performance.
- Length of text is tricky. Longer truncated text doesn’t always mean better performance.
- Large number of training samples present better performance.
- The model is very useful for the case of fake news detection in a subfield with limited training data.

Future Plan

- Training metrics should be label difference but not text difference.
- Compare the performance of different LLM model.
- It is interesting to check its performance on fake news generated deep fake