Fake News Detection by Supervised Fine Tuning of Tiny Llama2 Model

Introduction

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. The wide and fast spread of fake news online has posed real-world threats in critical domains like politics, economy, and public health.

There are organizations, like the House of Commons and the Crosscheck project, trying to deal with issues as confirming authors are accountable. However, their scope is so limited because they depend on human manual detection, in a globe with millions of articles either removed or being published every minute, this cannot be accountable or feasible manually. Automatic fake news detection, which aims at distinguishing inaccurate and intentionally misleading news items from others automatically, has been a promising solution in practice. Though much progress has been made, understanding and characterizing fake news is still challenging for current models. This is caused by the complexity of the news-faking process: Fake news creators might manipulate any part of the news, using diverse writing strategies and being driven by inscrutable underlying aims.

Textual content-based fake news detection methods are mainly dependent on the features extracted from the text that the classifier relies on in identifying fake news, such as linguistic features and syntactic features, sentiment features, or features based on the style and quality of the writing. Therefore, textual content-based method is promising for detection of fake news. Large language models (LLMs, which are usually trained on the larger-scale corpus and aligned with human preferences, have shown impressive emergent abilities on various tasks and are considered promising as general task solvers. It is interesting to build a LMM model to detect fake news and compare its performance comparing to machine learning classification models such as logistic regression, decision tree, random and forest, etc.

In this project, I built a fake news detection model by fine tuning TinyLlama with Lora (Low rank adaption). 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.

LoRA (Low-Rank Adaptation of Large Language Models) is a popular and lightweight training technique that significantly reduces the number of trainable parameters. It works by inserting a smaller number of new weights into the model and only these are trained. This makes training with LoRA much faster, memory-efficient, and produces smaller model weights (a few hundred MBs), which are easier to store and share.

I found true and fake news data set from https://www.kaggle.com/datasets/stevenpeutz/misinformation-fake-news-text-dataset-79k. The dataset containing 79k news, among which 34975 are 'true' articles and 43642 articles of misinfo, fake news or propaganda. The 'true' articles come from a variety of sources, such as Reuters, the New York TImes, the Washington Post and more. The 'fake' articles are sourced from several agencies. Firstly, American right wing extremist websites (such as Redflag Newsdesk, Beitbart, Truth Broadcast Network). Secondly, A previously made public dataset described in the article ”Ahmed H, Traore I, Saad S. (2017) “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127-138)”. Thirdly, Disinformation and propaganda cases collected by the EUvsDisinfo project. A project started in 2015 that identifies and fact checks disinformation cases originating from pro-Kremlin media that are spread across the EU.

Data exploration and preprocessing for traditional model.

There are two input files for true and fake news separately. There are 34975 articles in true news file and 43642 in fake news file. There are no other meta data for the dataset. I implemented standard text preprocessing on all the text data including transforming to lower case, removing punctuations, number, stop words and rare words, lemmatization, etc. After that I used tf-idf encoded the text data for the purpose of exploring the performance of traditional machine learning model’s performance.

Performance with traditional machine learning model

Using encoded tf-idf data as input, I checked the performance of logistic regression, decision tree, gradient boosting classifier and random forest on fake news detection. The data are split into train, test and valid by 70%, 20% and 10%. 5 fold cross validation is used to select the best performed parameters. After that, the results are compared based on validation dataset.

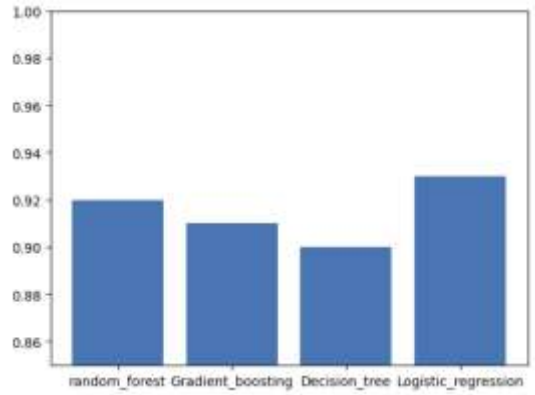

Figure 1. Baseline performance with traditional ML model

TinyLlama model generate prompt, train and validation

Considering easy control of the computation resource, I choose to train the model with limited number of samples, which I tried 1000, 2000 and 5000. 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

We can find that for the same number of samples and truncate size. Higher epochs give better performance. Therefore, we choose to use epoch3. 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. So longer text could be less helpful. We can also find that higher number of training samples present better performance. The best performance model is the one using 5000 training samples, epochs 3 and truncated text length of 256.

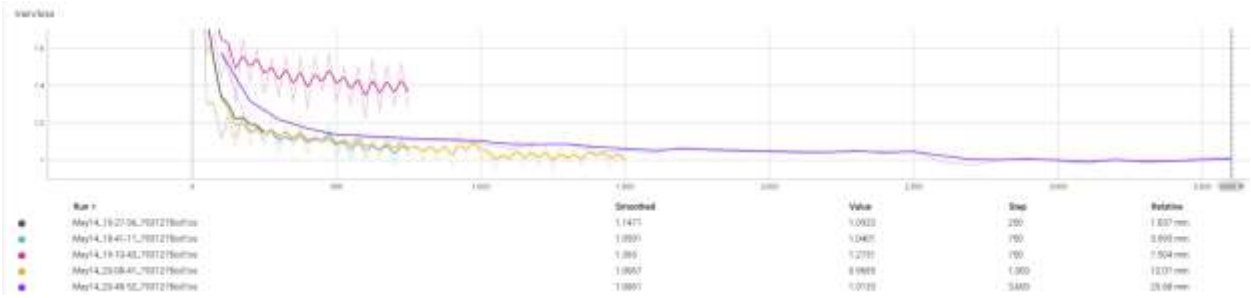

Figure 2.Training Loss wit TinyLlama from differnet parameters

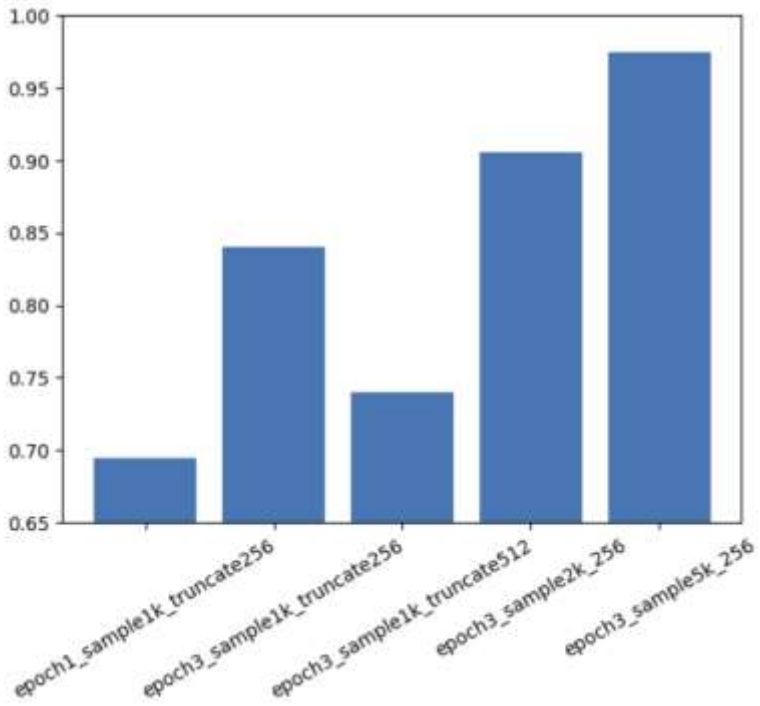

Figure 3. TinyLlama performance with different parameters.

Conclusion

TinyLlama can present comparable or better performance with limited number of training data. The model is very useful for the case of fake news detection in a field with limited 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. We should choose to use as many samples as possible based on available computation resources. In the future, I will further explore to have training metrics to be label difference but not text difference. I am also interesting to try different LLM models and have bench comparison on them. It is interesting to check its performance on fake news generated deepfake algorithm.