Leveraging Large Language Models for Fake News Detection

MSc Thesis Computer Science & Engineering

Merlijn Mac Gillavry

TU Delft

Leveraging Large Language Models for Fake News Detection

MSc Thesis Computer Science & Engineering

Thesis report

by

Merlijn Mac Gillavry

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on October 2 2024

Thesis committee:

Chair:	Professor Sole Pera
Supervisor:	Professor Sole Pera
Daily Supervisor:	Professor Sole Pera
External examiner:	Professor Luis Cruz
Place:	Faculty of Electrical Engineering, Mathematics, Computer Science, Delft
Project Duration:	From January 2024 - To October 2024
Student number:	4478363

Faculty of EEMCS · Delft University of Technology

TU Delft Delft
University of
Technology

Abstract

The spread of fake news has negatively impacted society. Prior efforts in Natural Language Processing (NLP) have employed machine learning models and Pre-trained Language Models (PLMs) like BERT to automate fake news detection with promising results. These models excel at text classification tasks, but their dependence on large context-specific datasets presents a hurdle in the dynamic and ever-evolving context of fake news. The recent emergence of Large Language Models (LLMs) offers a potentially transformative innovation. LLMs have demonstrated great promise in NLP tasks with little additional training data. Compared to PLMs, LLMs have broader knowledge and enhanced reasoning capabilities, suggesting their suitability for fake news detection. This study proposes an investigation which considers multiple perspectives to probe the applicability of the effectiveness, opportunities and challenges associated with leveraging LLMs for automated fake news detection. We leverage multiple state-of-the-art LMMs by using them at multiple levels of guidance to validate their accuracy and task-specific bias. The main contribution of this work is a better understanding of the role LLMs can play in automated fake news detection, the pitfalls that should be avoided when leveraging them, the most effective approaches when employing them, and the future challenges that need to be addressed.

Our key contributions include a better understanding of how to employ LLMs for fake news detection, their strengths and weaknesses, and some practical recommendations for deploying them in the real world. Our work shows that while LLMs hold great potential for advancing automated fake news detection, thoughtful consideration of their limitations and careful application refinement are essential for their effective deployment in the fight against fake news.

Preface

First, I would like to thank my thesis coordinator, Professor Sole Pera, for her continued help, support, and feedback. Our weekly sessions gave me meaningful insights, and her inquiry helped greatly. Second, I want to thank my friends and family for giving an outside perspective and for asking thought-provoking questions. I also want to thank Professor Luis Cruz for taking on the role of external examiner. Finally, I want to thank the Delft University of Technology for providing me with the formal education to become a master of science.

I specifically chose this topic because I wanted to conduct scientific research that was meaningful for society and helped overcome a difficult but important problem. As this research does not solve the problem of fake news in its entirety, I believe it can be an important stepping stone and a meaningful part of research in this domain.

Contents

List of Figures

List of Tables

Part I

Overview

1 Introduction

Since the widespread adoption of the internet, social media platforms, and digital news sources, it has become clear that the dissemination of misinformation/ fake news can pose significant threats. To illustrate fake news's effects on people, businesses, and countries, we give the following examples. During the COVID-19 pandemic, hundreds of Iranians died from drinking neat alcohol because of a fake story that a UK tabloid paper spread about how people cured themselves with whiskey and neat honey [1]. In the aftermath of the 2020 US presidential election, fake news was spread about the *Smartmatic* Voting machines used in Florida, resulting in serious reputational damage[1]. In Myanmar fake news spread through Facebook by influential people and government officials caused 750,000 people to flee the country[2] which was later determined to be a genocide and crime against humanity by the UN[3].

Fake news often exploits inherent cognitive biases, which are mental shortcuts that people use to process information quickly but can lead to judgment errors. One such bias is confirmation bias, where individuals are likelier to believe information that confirms their pre-existing beliefs and disregard information that contradicts them. This bias is particularly potent in the context of social media, where algorithms are designed to show users content that aligns with their preferences, reinforcing echo chambers. These echo chambers can, in turn, amplify the dissemination of fake news, leading to widespread misinformation that affects large segments of the population.

Historically, before the advent of the internet, the ability to disseminate information to most of society was predominantly in the hands of large organizations, such as media companies and government institutions. This gatekeeping role meant that information flow was relatively controlled and centralized. However, the introduction of the internet and, more specifically, the rise of Social Media Networks (SMNs) has dramatically democratized access to a large audience. Today, nearly anyone can reach many people with minimal effort, leading to an unprecedented spread of information and misinformation. Unfortunately, this newfound accessibility has been exploited by individuals and groups seeking to spread misinformation, often with harmful consequences.

As fake news proliferates and gains traction, distinguishing between genuine news and misinformation becomes increasingly challenging. This erosion of trust in news sources complicates the ability to discern truth from fiction. It can lead to a fragmented reality where a shared understanding of facts is no longer possible. Such a breakdown in shared reality poses far-reaching indirect consequences, including the undermining of democratic processes, increased polarization, and social unrest.

To combat this, various organizations have been established to fact-check and mitigate the impact of fake news. Traditionally, this has been achieved through objective journalism and the work of experts in the relevant fields, which was effective when only certain parties were in control of the news cycle. However, this process is labour-intensive and requires significant expertise, making it difficult to scale, especially since creating fake news is often much easier and faster than debunking it. In response, researchers have proposed automated fake news detection systems to address this challenge. With advancements in Natural Language Processing (NLP), automatically analyzing and classifying textual content became possible, thereby identifying potential fake news. However, these early approaches required vast amounts of specially curated data to achieve even modest effectiveness.

[1] https://apnews.com/article/election-2020-joe-biden-donald-trump-technology-electronic-voting-cd68ad2022611a36154ff3f243fcd1d8

Subsequent advancements in AI introduced Pre-trained Language Models (PLMs), such as Google's BERT, which are trained on enormous corpora of text and can be fine-tuned for specific tasks like text classification in fake news detection. These PLMs marked a significant improvement over earlier methods, as they provided a foundation upon which more accurate and adaptable fake news detection systems could be built. Nevertheless, these models are not without their challenges. Training and fine-tuning PLMs are expensive and computationally intensive, necessitating access to vast amounts of data and specialized hardware.

Current state-of-the-art methods rely on Machine Learning (ML) and Artificial Intelligence (AI) techniques that leverage PLMs, such as BERT [4], and often combine them with other ML systems that consider additional factors like the patterns of information spread and the profiles of those disseminating the information. While these approaches show considerable promise, they are still constrained by the significant resources they require and need much domain knowledge and expertise in ML technologies.

The rise of Large Language Models (LLMs) introduces new dynamics into this landscape; on the one hand, LLM-generated fake news can be disseminated on a larger scale and faster, making it even easier for malicious actors to manipulate public opinion. While tech companies are working to make these models safer and less prone to misuse, ethical considerations arise. One significant concern is determining who can decide what constitutes the ”truth.” Another critical issue is where to draw the line between making AI models safe and avoiding the censorship of free speech. Should AI companies be entrusted with the power to define truth? Can we trust that these models will not produce false or misleading information, known as *hallucinations*? These essential questions must be openly and freely discussed to ensure a safe and ethical future.

Conversely, LLMs also have the potential to serve as powerful tools for detecting and mitigating the spread of fake news. Just as these models can generate fake news, they can also be employed to identify it, providing a valuable resource in the fight against misinformation. However, defining their specific role in fake news detection is crucial to effectively leverage LLMs in this capacity. This requires ongoing research, collaboration, and dialogue among technologists, ethicists, policymakers, and the public.

Therefore, this manuscript investigates the role of LLMs in automated fake news detection. Specifically, our primary objective is to investigate the effectiveness, opportunities, and challenges of employing LLMs in this domain. To reach this objective, we aim to answer the following main research question: **How do Large Language Models fare when leveraging them for automated fake news detection?**. Before we answer this research question, we will first answer the following sub-questions:

Research Question 1

How does performance differ across widely used LLMs and associated prediction generation strategies?

Research Question 2

How does performance differ between LLMs and earlier automated fake news detection strategies, specifically feature-based and pre-trained language model strategies?

Research Question 3

How do LLMs fare in fake news detection from the different perspectives of the domain, textual characteristics, psychology, and sustainability?

To answer these questions, we adopt the following systematic approach. Our investigation begins by reviewing relevant background literature on the nature of fake news from both a web information perspective and a communication science perspective. Following this, we delve into the topic of fake news detection. This includes introducing different strategies used and challenges associated with them to explain and contextualize the problem we aim to solve. We then describe what LLMs are, how they work, and important considerations when using them, providing context for our evaluation of the models.

Next, we examine the most relevant and closely connected research that informs our work, providing an overview of previous studies, which we later use to contextualize our results. By identifying gaps and limitations in existing literature, we outline how our work builds on previous science and how it differs from earlier efforts to detect fake news automatically.

Following this, we present our methodology for evaluating LLMs in the context of fake news detection. Our evaluation framework incorporates several established datasets containing instances of fake and real news labelled on their veracity. These curated datasets represent a broad and diverse range of news articles, reflecting the variety of information consumed and disseminated online. This diversity is paramount for ensuring that our findings are robust and applicable across multiple domains of news consumption.

Additionally, we discuss our specific experimental setup for our work and the pipeline we developed using Python to ensure reproducibility. We then discuss ethical considerations, which are important for research on fake news and can have severe consequences when not considered responsibly and ethically.

To analyze the performance and architecture of LLMs, we select four widely used models developed by important players in the LLM space, chosen based on their architecture, openness, and popularity. These models include: **Gemma-2b-it**, **Mistral-0.2-7b-it**, **Llama-3.1-8b-it**, and **Gemma-2-9b-it**. These models are then tasked with predicting the veracity of news articles within the datasets using several distinct *detection strategies*. By comparing the behaviours and predictions of these LLMs over various detection strategies, we aim to answer sub-question 2. The detection strategies we employ vary in the levels of guidance provided to the LLMs for fake news detection. These include:

1. **Binary Classification**, where the models classify news as real or fake.
2. **Discrete Classification** offers more categories for classification, leaving a little more room for nuance in the predictions.
3. **Continous Classifications**, offering the most granular types of predictions from a scale of 0% to 100% true.
4. **Chain of Thought (CoT) Binary Classification**, which employs the models to reason about their classification.
5. Finally, we implement **Fine-tuning** on our models, specifically for **Binary Classification**, aiming to improve the base capabilities of the LLMs for fake news detection (available at huggingface.co[2].

We first evaluate the LLMs' raw performance by calculating the accuracy of the different detection strategies. We then explore their performance further by investigating their misclassification. We do this by finding the rate of false positives and false negatives. This tells us where the LLMs make more mistakes and how they make them. We also evaluate the validity of LLMs prediction labels. Since they are next token predictors, they can give semantically similar responses to classifying something as fake or real but are not the expected predictions for automatic parsing and processing. This evaluation of performance helps us answer question 1.

We then contextualize our findings with comparisons to baseline models introduced in earlier research. These baseline models help us answer question 2 as we explore whether and how LLMs provide tangible improvements over traditional methods for fake news detection.

Finally, we broaden our evaluation beyond the predictions' raw performance by analyzing the classifications of the different models through various lenses. These perspectives are grounded in earlier works and give a holistic overview of the challenges and qualities of leveraging LLMs for fake news detection. By evaluating the LLM predictions through these lenses, we answer question 3. These lenses include: **Textual Characteristics**, specifically text length and readability. **Psychology**, namely sentiment. **Domain**, based on the topics of the used news articles and finally **Sustainability**, measured in time and energy spent for inference and fine-tuning, to understand the impact on the environment of leveraging LLMs for fake news detection. All code to reproduce our experiments is available on Gitlab[3].

The results show that LLMs can outperform earlier baselines, but employing them presents crucial challenges. Although our work is thorough and expansive, continued work is needed to ensure the mitigation of widespread fake news propagation.

[2] https://github.com/Merlijnmacgillavry/FakeNewsDetectionUsingLLMs

[3] https://github.com/Merlijnmacgillavry/FakeNewsDetectionUsingLLMs

The key contributions of this study are outlined as follows. We aim to gain a better understanding of how to employ LLMs for the task of fake news detection, with practical recommendations. We also introduce a multi-perspective approach to contextualize and analyze LLMs for fake news detection. Finally, we discuss and share a better understanding of the trade-offs that need to be considered when using LLMs for fake news detection.

Part II

Background & Related Work

2

Literature Background

This chapter describes the research background regarding fake news detection and LLMs. We first give an overview of the scientific background of fake news, its impact, and the different kinds of fake news. Then, we discuss different existing automated fake news detection strategies. We start by describing manual fake news detection, then we discuss different automated techniques, and finally, we give an overview of the relevant background for LLMs.

2.1. Fake News, Misinformation and Disinformation

According to Lazer et al., fake news is defined as: *"Fabricated information that mimics news media content in form but not in organizational process or intent"* [5]. Fake news is strongly connected to other forms of false information sharing such *misinformation* (without the intent to deceive) and *disinformation* (with the intent to deceive and mislead people). Finally, there is the last form of information sharing that we categorize as fake news: *Malinformation*. Misinformation is true information taken out of context with the intent to harm. The main difference between fake news and real news is the responsibility taken, methodology, and objective approach that the sharing of real news follows in contrast to fake news. These journalistic norms of objectivity, responsibility, and multiple perspectives came to be after the widespread use of propaganda in World War I and, consequently, the distrust in news organizations because of their role in spreading it [5].

With the general public's introduction and widespread adoption of SMNs, fake news has become more easily accessible and shared. Additionally, SMNs are infected with automated social bots sharing, liking, and commenting on fake news. According to Facebook, it is a growing problem with Facebook estimating that as many as 60 million bots had been occupying their platform in a U.S. senate hearing[1] and *X* (formally *Twitter*) estimating that between 9% and 15% of active accounts are bots [6]. It is hard to mitigate this problem of bots and bad actors on SMNs. The vast amounts of news spread over many different news sources make it hard for the average person to distinguish between real and fake information.

Previous works have tried to identify the leading causes of fake news and how to mitigate it. One such example is the work of Chen et al. [7], in which they did a systematic literature review on the key factors contributing to the spread of misinformation through the four components of the SMCR [8] model introduced by David Berlo in 1960. The SMCR model divides communication into four components: source, message, channel, and receiver. Chen et al. further divide the different components into smaller ones:

- Message-related factors
  - Presentation formats
  - Language styles
  - Psychological cues
- Source factors
  - Credibility

[1] https://www.judiciary.senate.gov/committee-activity/hearings/extremist-content-and-russian-disinformation-online-working-with-tech-to-find-solutions

  - Ambiguity
  - Popularity
  - Similarity
  - Self-interest
- Context-related factors
  - Normative cues
  - Channel attributes
  - Social network features
  - Debunking activities
- Receiver factors
  - Demographics
  - Personality
  - Worldview
  - Cognition
  - Motivation
  - Emotion

These smaller components can help systematically approach research into fake news and gain a more holistic understanding of the spread of misinformation.

Not all fake news is created equally; fake news can be mostly unsubstantiated rumors to gain clicks on websites, political propaganda with the intent to sway elections, or news about science made with the intent to help and inform people but without the scientific know-how to understand the relevant literature. These different types of fake news are not all as easy to detect and require different amounts of domain knowledge, intent, and reputation of the source and knowledge of current events. Because of the variety in content, presentation, and topics of different forms of fake news touch, accurate detection of fake news is everything but a simple problem to overcome.

2.2. Fake News Detection

To mitigate fake news, it first needs to be identified. In this section, we describe the background of different forms of fake news detection.

2.2.1. Manual Fake News Detection

Fake news detection has historically been a manually performed and labour-intensive process. The manual detection of fake news often involves different roles, such as a domain expert, someone specialized in domain news, such as an experienced journalist in the field, and multiple reviewers. Different institutions and organizations have aimed to detect and fact-check fake news. Nonetheless, this is not a profitable or glamorous endeavour, making the number of organizations small and often underfunded. Spreading fake news, however, can have clear monetary and reputational incentives [9].

One such organization that fact-checks news is *PolitiFact*. PolitiFact uses a standardized process of fact-checking news. This process involves one reporter and 3 editors who discuss the following four questions[2]:

1. Is the statement literally true?
2. Is there another way to read the statement? Is the statement open to interpretation
3. Did the speaker provide evidence? Did the speaker prove the statement to be true?
4. How have we handled similar statements in the past? What is PolitiFact's jurisprudence?

This manual effort is not a sustainable and scalable approach to mitigating fake news, as a vast amount of

[2] https://www.politifact.com/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-i/

fake news is shared and produced online.

Another approach to manual fake news detection is to employ the users of the SMNs on which the news is shared by leveraging crowdsourcing. The most famous example of fact-checking through crowdsourcing is Wikipedia[3]. Another well-known SMN that does this is X.com (formerly Twitter), using Community Notes. However, this approach has issues: Crowdsourced fact-checking generally relies on a few users contributing most of the work. For example, in a study about contributions on Wikipedia [10], Ortega et al. found that less than 10% of the authors were responsible for over 90% of the contributions. Additionally, as with normal news creation, there is no guarantee that bad actors do not do the crowd-sourced notes and fact-checking.

2.2.2. Automated Fake News Detection

Another approach to fake news detection is employing algorithms and AI models. This method is easier to scale than manual fake news detection and does not need to be investigated in terms of incentives to counteract bad actors. In short, automated fake news detection strategies use programmed systems that are trained on a dataset of labelled news and then can automatically classify the veracity of the news based on the patterns it learned from the training data. In this work, we divide the different kinds of automated detection into feature-based and PLM detection strategies. Depending on the perspective taken and the context in which they are discussed, there are many ways to categorize different ML and AI technologies. Our division into feature-based and PLM comes from a historical and characteristics-based approach. Feature-based classification strategies have been around longer than PLM-based methods but are often less effective in NLP.

Feature-Based Feature-based fake news detection models extract various attributes from news articles, which can then be used to train ML models to classify news as either fake or real. As with all feature-based AI models, it often takes domain knowledge and understanding of NLP techniques to know which features are helpful for training the models and which are not. Researchers in fake news detection can ground their choice of features on the SMCR model. These features can include *Textual features* such as Linguistic Inquiry and Word Count (LIWC), *Psychological features* such as the sentiment of the text. This kind of feature-based strategy can also be combined with non-message-related factors such as the reputation of the accounts sharing news on an SMN. Generally, the process of leveraging feature-based models for fake news detection can be summarized by the following steps:

1. Extract relevant features: Features are extracted from the news article, including linguistic features (e.g., word frequency, complexity, sentiment), stylistic features, and source-related features (e.g., author reputation, social media metadata).
2. Train an ML algorithm: The extracted features are used to train a machine learning algorithm such as Support Vector Machine (SVM), Naive Bayes, or Random Forest on a labelled dataset of news articles. The choice of algorithm may depend on the type of features and the amount of data available.
3. Classify New articles: Once the model is trained, it can be used to classify new, unseen articles based on the same set of features, predicting the veracity of the news.

Provided the features are well chosen, and the dataset these models are trained on is diverse and a good representation of the task at hand. Their ability to differentiate between fake and real news typically improves. There are, however, downsides to this approach. Extracting the correct features is not an exact science. Often, experience in developing these models helps a lot with choosing the right ones. There are ways to validate extracted features, but there is no guarantee that they will be optimal. Additionally, while these models perform well in specific contexts, they are not guaranteed to work as well on unseen data or data that was underrepresented in the training set.

Pre-trained Language Models Another automated fake news detection approach is using PLMs such as BERT [4]. PLMs differ from feature-based technologies because they do not need manually extracted features. They use self-supervised learning methods such as a masked language model on a huge text corpus to “understand” natural language. They are typically based on transformer architectures and are more challenging to interpret than feature-based models because of the lack of clarity in these models’

[3] https://en.wikipedia.org/wiki/Wikipedia:About

internal representation of language. Their language representation could also be described as latent features that capture nuances of word meaning, sentence structure, and other linguistic patterns. These PLMs can be easily adapted to downstream tasks such as fake news detection by fine-tuning them to a specific dataset. They then use their already-learned understanding of natural language and the newly trained context to adapt to the task. What makes PLMs such as BERT strong is that they can understand text in the context in which they are situated. Typically, the process for using a Pre-trained Language Model for fake news detection is to:

1. Pick or Pre-train a Language Model: Start with an existing pre-trained language model or pre-train a new model from scratch on a large corpus of text. The pre-training allows the model to learn general language patterns such as grammar, syntax, and semantics.
2. Fine-tune the Pre-trained Model on Labeled Data: Fine-tune, the pre-trained model on a dataset specifically designed for fake news detection. This dataset is typically labelled with ”ground truth” (i.e., articles that are clearly identified as fake or real). Fine-tuning allows the model to adjust its language representations to the task at hand, learning the specific patterns that distinguish fake news from real news.
3. Classify Fake News: Once fine-tuned, the model can be used to classify new, unseen news articles. The model can predict whether an article is fake or real by leveraging its pre-trained language understanding and the patterns it learned during fine-tuning.

PLMs can be great for tasks such as fake news detection because they understand the context. However, there are also challenges related to using PLMs; one of them is that they lack interpretability compared to feature-based strategies. Additionally, they also require more computing resources than feature-based strategies.

2.3. Large Language Models

Recent developments in the realm of AI and Machine Learning have produced a new phenomenon: LLMs. One way to think of them is as spiritual descendants of the smaller and older PLMs. Although their architecture is often not the same, LLMs, for example, mostly use decoder strategies, while PLMs can use encoder-decoder architectures. They also share many similarities regarding how to use and adapt them. These models are characterized by their enormous training datasets and billions of parameters. Because of the large datasets and the tremendous amount of computing power needed, most LLMs are made by large tech companies. How these companies gather their training data and how they specifically train their LLMs is generally hard to find out. Contrary to what these companies' names suggest, this information is not shared openly, and therefore, these models can mostly only be investigated as black boxes without knowing their inner workings.

Models Choosing the right model for a specific task can be hard and can often only be done by comparing the outputs generated by these models given the same input. There are, however, some characteristics that can hint at the LLMs' performance. One of these characteristics is their parameter size. The parameter size, typically just referred to as the size of a model, represents the amount of weights that the model uses to fit complex patterns into the data on which it was trained. Based on the input tokens a model gets as input, it then predicts a sequence of tokens as output by a computation using the weights of the model that has the highest probability of coming next and thus predicts which sequence of text is most likely to follow after the input [11]. Generally, a higher parameter size means a better-performing model. Some of these companies and organizations choose to fine-tune their models on specific tasks. These models are often trained with human feedback to provide more fitting and, therefore, better output. Many popular LLMs like LLama, Gemma, and Mistral have, for example, *Instruction Tuned* versions that generally perform better on human instructions.

Prompting Prompting in the context of LLMs is the textual input users give to guide the model's output [11]. A new discipline in generative AI is the practice of *Prompt Engineering*, where different prompting techniques are compared to better understand how to most effectively guide the LLMs to better output. Prompt engineering has introduced new techniques such as *Chain of Thought* (CoT) prompting [12]. CoT prompting tends to give better results for specific instructions that require some kind of reasoning, such as basic math, by prompting the LLM to perform the task and sub-tasks that need to reason through how they perform their main task. Other prompt engineering techniques include: *Tree of Thought* (ToT) prompting to evaluate multiple different solutions and reasoning steps to conclude [13], *Self-Consistency* to investigate whether LLMs are consistent in their output by generating multiple solutions to the same input and checking their similarity [14], *Reflection* to make LLMs reflect on their own output for correctness [15], *Expert Prompting* which simulates multiple experts in the fields pertaining to the prompt which is then combined to give a correct output [16], *Chains* for combining the capabilities of multiple LLMs in a sequential matter to perform difficult tasks [17], *Rails* to guide and control the output of LLMs by predefined rule sets and finally *Automatic Prompt Engineering* using LLMs to generate prompts to use for LLMs for specific instructions [18]

Decoding strategies LLMs can be guided to higher probability output by choosing the appropriate decoding strategy. These decoding strategies determine how the LLM “chooses” the next token to generate. Some popular strategies are *Greedy Search*, taking the most probable token and discarding the other options and *Beam Search*, keeping the **N** most likely sentences (sequences of tokens) at each step during decoding, and in the end selects the generated response with the highest probability, the sentences used in beam search can then also be normalized according to sentence length to mitigate the issue of beam search preferring shorter sentences. When using LLMs to generate output based on different inputs, hyperparameters can be chosen to guide the LLM to preferable output. These include parameters like temperature, a value between 0 and 1.0 that determines the ”creativity” of the output [11]; using higher temperature models can give diverse output, making it more akin to how humans would answer questions slightly differently if they were asked the similar question over and over again when the temperature is set to 0 that is equivalent to greedy search.

Fine-tuning As with PLMs, deployed LLMs can be further trained by providing additional training data to improve them at downstream tasks. This process of fine-tuning mainly consists of two parts:

1. *Instruction tuning*: To better specific abilities of LLMs, like following the input instructions for a specific task.
2. *Alignment tuning*: To align the models with human preferences and values.

To contextualize the two steps of fine-tuning with our work, you could view instruction tuning as training the models to give us usable answers such as ”True” and ”False” and alignment-tuning as training the model to classify the news with the correct label. There exist different forms of fine-tuning. Full fine-tuning, for example, recalculates all/most of the model parameters for a specific task. This can, however, be very costly and time-consuming as these models have giant amounts of parameters. To mitigate these issues, techniques such as Parameter-Efficient Fine-Tuning (PEFT) [19] fine-tuning have been proposed. These techniques only fine-tune a subset of the parameters of a model for downstream tasks, greatly reducing the needed computing and time. One of these PEFT techniques is Low-Rank Adaptation (LoRA) [20], which is based on the insight that instead of updating all weights in a large model, one can represent the weight changes in a low-rank decomposition. These weight changes can then be used as an adapter for the original weights to get fine-tuned results.

Sustainability With the amount of models being trained on vast corpora of data and the parallel growing concerns of climate change, we cannot discuss these LLMs without also discussing their impact on the environment. The largest models are made by *BigTech* companies such as OpenAI, Meta, and Google and often use a vast amount of specialized hardware. How these companies share the amount of impact on the environment their models have differs per instance. Google, for example, discloses the hardware architecture it uses and gives an estimate of how many tons of CO2 it produces by training its models [21, 22]. On the contrary, Meta does give some information about the hardware and some insight about the training time but does not give a useful metric for impact on the environment, but ironically does mention that the environment has an impact on their model training because of fluctuation in temperature. Generally speaking, smaller models take less computational resources and less time to train, making them better for the environment.

Open LLMs The different companies creating LLMs show great variation in the level of “openness” of their model. Some organizations adopt a fully open approach where model architecture, training data, weights, and training techniques are shared publicly [23, 24]. Others may share the model weights and provide a general overview of the training data. Still, they are less transparent about their training techniques and the specific datasets they used [25, 26, 27, 22, 21]. Finally, some models are accessible only through proprietary APIs, with no public disclosure of their underlying components [28].

When selecting an LLM for scientific investigation, particularly in a research setting, it is crucial to consider the level of openness. On the one hand, using fully open models allows for a more thorough examination of confounding factors such as data contamination. On the other hand, incorporating more widely used, albeit less open, models can enhance the real-world applicability of the research findings.

3 Related Work

This section discusses the works most closely related to our investigation. We discuss relevant feature-based technologies and PLMs used earlier in the context of fake news detection. Then, we discuss some of the state-of-the-art work that has been done with LLMs for text classification, specifically fake news detection.

3.1. Feature Based Technologies

As described in the previous chapter, earlier research has used feature-based technologies for automated fake news detection. These feature-based technologies use varying NLP and AI techniques to detect fake news. We first describe work done on message-related factors of the SMCR model and then discuss work done on non-message-related factors.

3.1.1. Message Related Factors

One of the more straightforward methods was introduced by Granik et al. [29], who applied a Naive Bayes Classifier (NBC) to classify fake news based on the frequency of words in labelled fake and real news articles. Their NBC achieved an accuracy of 75.4%, which is notable for its simplicity. However, this result should be interpreted with caution. The dataset used in their study was highly imbalanced, with only about 5% of the articles being labelled as fake news. This skewed distribution limits the generalizability of their findings, as the classifier may not perform as well on a more balanced dataset or in real-world scenarios where the proportion of fake news varies. Therefore, while their work highlights the potential of feature-based technologies in fake news detection, it is important not to draw definitive conclusions about the effectiveness of NBC in this context. Instead, their study underscored the need for further exploration and validation of feature-based methods in diverse and balanced datasets.

Around the same time, Horne and Adali [30] used a similarly skewed dataset, but they employed a much broader range of textual features for fake news detection. They focused on three types of features:

- **Stylistic Features**: These include elements such as the frequency of stopwords, punctuation, and the use of swearwords.
- **Complexity Features**: These capture the overall complexity of the content, measured by well-known readability indexes like the Gunning Fog, Dale Chall score, SMOG Grade, and Flesch-Kincaid Grade Level indexes.
- **Psychological Features**: These are designed to assess the sentiment of the text.

Their study was mainly focused on investigating whether either the title or the body of an article was a better predictor for the veracity of a news article. They found that using a Support Vector Machine (SVM) on the titles of their articles, they got an impressive accuracy of 78% compared to the 71% they got for the body. This work showed that even basic content-based approaches have potential in terms of accurately detecting fake news.

To extend this work, Shrestha et al. conducted a reproducibility study [31] that incorporated additional features and a more comprehensive dataset, including data from PolitiFact and GossipCop. In their study, they not only examined the Support Vector Machine (SVM) used by Horne and Adalı but also evaluated a Random Forest (RF) classifier and a Linear Regression classifier. Their findings indicated that for the BuzzFeed dataset used by Horne and Adalı, the results were consistent with the original study. However, when applied to the PolitiFact dataset, the RF classifier outperformed the SVM, particularly when analyzing the body of the news articles rather than the titles, achieving an accuracy of 87%. This reproducibility study underscores the variability in classifier performance across different datasets, highlighting the importance of context and data characteristics in the effectiveness of fake news detection models.

Another feature based approach by Dun et al. [32], was to extract entities from a dataset of news articles to make a Knowledge-aware Attention Network for fake news detection. Dun et al. combined the general knowledge about entities found in news to detect fake news. Their work provided insight into how general knowledge about topics can contribute to accurately detecting fake news.

3.1.2. Non Message Related Factors

Another way to use these feature-based technologies is to not only look at the message-related features but also investigate how fake news is propagated through the network, who/what organization shares the news, and what kind of people are most likely to believe it. In other words, these techniques investigate the other parts of the SMCR model.

One such work is done by Cui et al. [33], where they used a Meta-path-based approach to both take news and how it is propagated through social media networks into account. They made a heterogeneous graph of news and user nodes to detect fake news and found a promising accuracy of 90% on a dataset containing around 2000 news articles comprised of the FANG [34] and the FakeHealth [35] dataset. Although their results are impressive, they use a relatively small dataset, which is not that diverse.

A similar work proposed by Min et al. [36] also used the network propagation information of fake news to create a model for fake news detection. Their approach used a Divide-and-Conquer strategy to combine the information of the news content and the post-user, post-post, and user-user graphs. They used a larger dataset with more than 27000 news events and tested their strategy with in-topic and out-of-topic split and reached an Area Under Curve (AUC) score of 91% and 64%, respectively. Their research shows that different topics have different characteristics that deep learning methods can leverage.

3.2. Pretrained Language Models

Other strategies for fake news detection used PLMs trained explicitly on a large corpus of text, which were then either used as-is or fine-tuned for fake news detection.

One of the works leveraging these PLMs was Wu et al. [37]. They proposed a few-shot fake news detection approach that combined the prompting of BERT [4] and veracity-guided social alignment. overcame the problem of label scarcity by using the inherent pre-trained knowledge of BERT and aligning that with social context for fake news classification [37]. The work of Wu et al. shows that PLMs can be leveraged for fake news detection, especially in combination with source and receiver factors of the SMCR model.

Another work by Jiang et al. showed that PLMs such as BERT can be combined with prompt learning to detect fake news. Jiang et al. showed that, given the right prompt structure and the already existing “general knowledge” in PLMs, they could be used to detect fake news better. This bolsters the capabilities of PLMs for fake news detection and shows that PLMs can be combined with feature-based technologies for even better classifications.

Whitehouse et al. [38] used the base understanding of the natural language of PLMs, in combination with knowledge bases, to improve their fake news detection capabilities. They found that on a dataset on Covid-19, they reached an incredible accuracy of 94%. There were, however, clear confounding factors making it much easier for the models to detect fake news. Their models recognized real news by the “https” string that occurred in 93% of the real news and only in 43% of the fake news. When evaluated on another dataset, their models reached an accuracy of around 28% . Whitehouse et al. show that PLMs can be combined with external knowledge sources (context factors of the SMCR model) to further bolster their ability to classify the veracity of news.

Szczepański et al. [39] investigated different methods on how to best use explainable AI (xAI) techniques to mitigate the black-box problem that arises when leveraging PLMs for fake news detection. This problem refers to the fact that even if these language models accurately detect fake news, it is hard to understand why they did. They found that with a diverse set of techniques, reasonable explanations were found for PLM-based classifications while minimizing architecture changes. This highlights that even though it is hard to validate the results of PLM-based fake news detection systems, they can be explained by using xAi techniques.

The works above show that PLM models can be effectively leveraged for fake news detection with impressive results. Especially, when combined with other factors of the SMCR model.

3.3. LLMs For Fake News Detection

Even though LLMs are generally used for generating content and predicting the most likely text that would follow input text. Earlier works have shown that LLMs can also be leveraged for text classification. For example, Liu et al. [40] used LLMs for binary stance classification in the realm of political science. Sun et al. [41] found that even with little domain-specific training data, the inherent generalization ability of LLMs could be leveraged for good text classification results. Even more impressively, Zhang et al. [42] found that in some cases, LLM-based text classification could even surpass humans in text classification tasks. These works show that there is a potential for LLMs in the realm of text classification, making it also an interesting alley to explore when trying to detect fake news automatically.

With the rise of LLMs, new research has emerged and continues to try leveraging this technological innovation in domains tackled by earlier AI and NLP strategies. As of the writing of this work, similar works are being proposed and worked on leveraging LLMs for fake news detection.

Liu et al. use an ensemble approach with different LLMs working together with outside information to determine the veracity of fake news [43]. Liu et al. used the FakeNewsNet datasets [44] and reached impressive 80%+ results with their approach. This work helps contextualize our work. Although impressive, it is less comprehensive than our work because we also take a multi-perspective lens approach and dive deeper into how to leverage LLMs just for content-based classification.

Another interesting work by Hu et al. uses LLM rationales with PLMs to help predict the veracity of news. They reached results between 70% and 80% using this approach, showing that the rationale and reasoning capabilities of LLMs can be leveraged to help detect fake news.

In contrast to the works described above, we do not explicitly introduce a new framework or model to detect fake news. Our work investigates how different LLMs, prompting techniques, and their fine-tuning differ when approached through several lenses, giving a reproducible and extensible overview of how LLMs can be leveraged for fake news detection.

Part III

Experimental Setup

4 Methodology

In this section, we outline the methodology employed to explore the potential role of LLMs in Fake News Detection. We begin by detailing the datasets utilized for evaluating the LLMs. Next, we describe the various models selected for our investigation and discuss the prompting techniques reviewed. We then explain how fine-tuning was used to push the LLMs to their limits. Finally, we present the different perspectives considered to provide a comprehensive understanding of the role LLMs can play, extending beyond overall performance metrics.

4.1. Datasets

To evaluate the classifications of the LLMs, it was essential to curate a diverse dataset comprising fake and real news articles. Acquiring datasets that are both high in quality—meaning fact-checked and sourced from trusted resources—and large enough for robust statistical analysis is challenging. Therefore, we combined three previously used datasets in related research to create a comprehensive dataset for evaluating LLMs in the context of fake news detection.

Our objective was to compile a diverse collection of labelled news articles spanning various topics that accurately represent real-life news categories commonly affected by fake news. As with many applications of AI and ML, using reliable and established data to evaluate our models was paramount. As a common mantra in ML goes: ”Garbage in, Garbage out”. Therefore, our approach was to find data that was used in relevant literature and had diverse characteristics. These categories include *Entertainment* and gossip news, *Political* news, and *Health* related news. The combined dataset consists of approximately 32,000 articles, varying in text length and topic. In this section, we will describe the selected datasets and their key characteristics.

FakeNewsNet In 2018, Shu et al. introduced FakeNewsNet [44], a dataset collected from Politifact and GossipCop. Politifact is a free fact-checking website primarily focused on political news. It employs a thorough and transparent methodology to fact-check claims made by or about politicians. The fact-checking process involves answering the described in Section 2.2.1. By addressing these questions, Politifact classifies each article into one of six categories, ranging from ”Pants on Fire” (for statements that are egregiously false) to ”True” (for statements that are accurate). Additionally, Politifact partners with Meta and TikTok to help curb the spread of misinformation online, further establishing its credibility in the realm of automated fake news detection.

GossipCop, on the other hand, was a website dedicated to fact-checking celebrity news. It rated articles on a scale from 0 to 10, with 0 representing completely false information and 10 indicating fully accurate news. FakeNewsNet comprises approximately 21,000 entries of news stories, each labelled as either fake or true.

The characteristics of the used FakeNewsNet dataset are detailed in Table 4.1

Partition	Count	Real Count	Fake Count	Real percentage	Fake percentage
Total	14,398	10,455	3,943	72.61%	27.29%
PolitiFact	606	311	295	51.32%	48.68%
GossipCop	13,792	10,144	3,648	73.55%	26.45%

Table 4.1: Count and ground truth characteristics of the used FakeNewsNet Dataset

FakeHealth The FakeHealth dataset [35] focuses on health-related misinformation, comprising two primary components: HealthRelease and HealthStory. HealthRelease includes content published by institutions, universities, and companies, while HealthStory contains news articles from media outlets such as Reuters Health. The dataset was reviewed and labelled by the now-defunct HealthNewsReview[1], a platform founded by Gary Schwitzer, a journalist with over four decades of experience in health reporting.

FakeHealth consists of 2,296 news stories, each rated on a scale from 1 to 5, with any rating below 3 classified as fake. In addition to the ratings, FakeHealth provides reasoning and explanations for each entry's rating based on 10 selection criteria reviewed by individuals with medical expertise:

- Does the news release adequately discuss the costs of the intervention?
- Does the news release adequately quantify the benefits of the treatment/test/product/procedure?
- Does the news release adequately explain or quantify the harms of the intervention?
- Does the news release seem to grasp the quality of the evidence?
- Does the news release commit disease-mongering?
- Does the news release identify funding sources and disclose conflicts of interest?
- Does the news release compare the new approach with existing alternatives?
- Does the news release establish the availability of the treatment/test/product/procedure?
- Does the news release establish the true novelty of the approach?
- Does the news release include unjustifiable, sensational language, including in the quotes of researchers?

These criteria offer valuable insights into the rationale behind the classification of health-related news as either fake or real. Similar to the approach taken with the FakeNewsNet dataset, we removed unusable data from FakeHealth to ensure the dataset's quality and relevance. The characteristics of the refined dataset are detailed in Table4.2

Partition	Count	Real Count	Fake Count	Real percentage	Fake percentage
Total	2155	1,429	726	66.31%	33.70%
HealthRelease	594	308	286	51.85%	48.15%
HealthStory	1561	1,121	440	71.81%	28.19%

Table 4.2: Count and ground truth characteristics of the used HealthStory Dataset

MOCHEG The Multimodal Fact-Checking and Explanation Generation (MOCHEG) dataset [45] is a multi-modal source collection of claims fact-checked by *Snopes* and *PolitiFact*. MOCHEG contains around 16,000 labelled claims with explanations either labelled true or false. MOCHEG was specifically curated for multimodal fact-checking and, therefore, contained many explanations or fact-checks based on videos and images. In this sense, MOCHEG differs from FakeNewsNet and FakeHealth because they were mostly fact-checked based on content. However, we still used this dataset to investigate how LLMs performed with fake news with little textual context. The characteristics of MOCHEG are displayed in Table 4.3

[1] https://healthnewsreview.org

Partition	Count	Real Count	Fake Count	Real percentage	Fake percentage
MOCHEG	15524	9670	5854	62.29%	27.71%

Table 4.3: Count and ground truth characteristics of the used MOCHEG Dataset

4.2. Models

In our evaluation, we opted for 3 different model architectures. These are feature-based and PLM-based fake news classifiers and LLMs, of which the first two act as baselines to compare with our main investigation into LLMs. As LLMs have become increasingly prevalent, numerous organizations and companies have developed their own versions. To gain a comprehensive understanding of the current LLM landscape, we selected four “open” models from Google, Meta, and Mistral. This selection represents a broad spectrum of contemporary LLMs, reflecting the diversity in model authorship and size. All models are available for download on HuggingFace[2]. We opted for openly available models rather than proprietary ones to ensure reproducibility and consistency in our and future evaluations.

4.2.1. Baselines

To contextualize the potential of LLMs in the task of fake news detection, we compare their performance against earlier automated content-based detection strategies. For this purpose, we have selected two distinct baseline models. These baselines serve as reference points to evaluate and contrast the effectiveness of the LLMs in detecting fake news, helping us answer research question 2.

Feature-Based Baseline The first baseline model we selected is based on the reproducibility study by Spezanno et al. [46], which investigated the claim by Horne and Adali [47]. Horne and Adali proposed that textual features from the title of an article are a better predictor of its veracity than features from the content itself. Spezanno et al. confirmed that this claim held true when using linear classifiers but found the inverse to be true with classifiers such as Random Forest. For our feature-based baseline, we adopted the methodology and code from Spezanno et al. [46] and applied it to our combined dataset. We specifically employed their Random Forest classifier, as it demonstrated the best performance in their study, to compare and contextualize the predictions made by the LLMs.

BERT-Based Baseline The second baseline model employed in our study is a BERT-based PLM prompting classification strategy, as outlined in the paper Prompt and Align by Wu et al. [37]. This method utilizes BERT, a smaller pre-trained language model, to classify the veracity of news articles. Specifically, the approach involves designing prompts to guide BERT’s classification of news content into categories of true or false. BERT was selected for this baseline due to its well-established effectiveness in understanding and processing textual information, making it a relevant benchmark for evaluating the performance of larger language models.

By comparing the results from BERT with those from larger models, we aim to assess whether the increased capacity of more recent models and their new architectures leads to improved accuracy for fake news detection. This comparison will help determine if advancements in model size and complexity contribute significantly to performance improvements in this task.

4.2.2. LLMs

To evaluate the capabilities of large language models (LLMs) in fake news detection, our study includes a diverse selection of state-of-the-art models from leading organizations and a diverse set of detection strategies (sometimes also called detection methods) to leverage them.

Architectures

We aim to assess how different model sizes, generations and architectures impact their capabilities for Fake News Detection and compare newer models against established benchmarks. For all selected models, we chose the *Instruction Tuned* versions due to their pre-training on instruction-based tasks, which simplifies the prediction process. An overview of our selected models with their release date, amount of

[2] https://huggingface.co/models

parameters and publisher is diplayed in table 4.4. By comparing and contrasting these various models we answer 1.

Model	Release Date	#Parameters	Publisher
Gemma-2b	April 2024	2 Billion	Google
Gemma-2-9B	June 2024	9 Billion	Google
LLama-3.1-8b	July 2024	8 Billion	Meta
Mistral-0.2-7B	December 2023	7 Billion	MistralAi

Table 4.4: Release date, size and publisher of used models

Google Gemma Models In our study, we evaluated two versions from Google's Gemma family of LLMs [21]: the 2-billion-parameter model (Gemma-2b) and the 9-billion-parameter model with a revised architecture (Gemma-2-9b) [21]. The Gemma-2b, as the smallest model in our evaluation, was selected to determine whether smaller, more accessible models can still effectively perform fake news detection when deployed on standard computing systems. This model is particularly noteworthy for its cost-effectiveness and suitability for organizations with limited computational resources. In contrast, the Gemma-2-9b, being the largest model in our comparison, enables us to explore the impact of larger model sizes and advanced architectures on performance. By including both models from the same family, we aim to gain insights into how model size and architectural advancements influence classification accuracy and to assess whether these improvements translate into better detection of news veracity.

Meta LLama3.1-8B To explore the capabilities of state-of-the-art large language models (LLMs), we included the latest model from Meta's LLaMa3 family: the LLaMa3.1-8B [27]. As the newest and reportedly best-performing model in the LLaMa series, it provides a relevant benchmark for comparison with the Mistral and Gemma models in our evaluation. The LLama family of models is highly popular in open-source and self-hosted LLM communities. It is powered by training data from one of the biggest SMNs in the world, namely Facebook, and is particularly interesting because of its state-of-the-art performance on common LLM benchmarks [26, 27].

Mistral-0.2-7B The Mistral-0.2-7B model has demonstrated notable performance in various benchmarks, according to Mistral's report [25]. This model is reported to surpass both the Gemma and LLaMa2 models in specific evaluations. Interestingly, Mistral's technical documentation claims that the Mistral-0.2-7B model outperforms even the larger LLaMa2 models, including those with 13 billion and 37 billion parameters. This performance assertion highlights the model's efficiency and effectiveness despite its relatively smaller size compared to some of its competitors. By including Mistral-0.2-7B in our study, we aim to assess its performance in fake news detection and verify Mistral's claims regarding its competitive edge over larger models. It is especially interesting because it is the only one of the evaluated LLMs we chose that an earlier established Big-tech company did not develop. The evaluation of Mistral also determines the possibility of newer vendors playing a part in the LLM ecosystem for downstream veracity classification tasks.

Detection Strategies
Given the diversity in rating systems used by different fact-checking organizations, as discussed in our dataset section, and the proven benefits of Chain of Thought (CoT) prompting and fine-tuning in enhancing LLM performance, we adopted a multi-tiered approach for prompting and investigating the LLMs. This multi-tier approach emulates different levels of guidance given to the LLMs for the task of fake news detection. We began by assessing the models in their *As-is* state, querying them with various output labels. Next, we applied CoT prompting to stimulate the models' reasoning abilities. Finally, we fine-tuned the models specifically for fake news detection to determine if their performance could be further optimized.

As-Is Evaluation: This initial assessment employs the LLMs in their native state, without any specific training for fake news detection. Before this evaluation, we conducted an exploratory phase to identify a suitable, effective prompting format. We compared the performance of the LLMs using three different query output formats:

Binary Classification The simplest and most directly comparable format to our dataset labels is binary classification. We instruct the LLM to classify a given article as either “fake” or “real” using the following prompt:

```
Instruction:
You are a binary news veracity classifier.
Given an article, classify whether it is real or fake.
Answer with "REAL" if it is real and "FAKE" if it is fake.
Article:
{ARTICLE}
Classification:
```

Binary Classification should be the easiest NLP task for the LLMs in that it only requires one of two answers. It most closely mirrors the behaviour we expect from the feature-based baseline, making it an interesting comparison.

Discrete Classification To mirror the rating system used by PolitiFact, we implemented a prompting strategy with their specific labels. The LLM is instructed to classify the article into one of six categories: Pants on Fire, False, Mostly False, Half True, Mostly True, or True. The prompt used is:

```
Instruction:
You are a discrete news veracity classifier.
Given an article, classify whether it is real or fake
by choosing one of the following options (from Fake to Real):
["Pants on Fire", "False", "Mostly False", "Half True", "Mostly True", "True"].
Article:
{ARTICLE}
Classification:
```

Discrete classification is the detection strategy that most closely mirrors how political journalists would label their manual classification. We would also expect good results from this if the LLMs are trained on the PolitiFact dataset. However, we cannot know for certain as the exact data on which these models are trained is not publicly available. This detection strategy also gives more room for nuance in the model’s classification than the Binary Classification strategy.

Continuous Classification As a complementary approach, we asked the LLM to act as a percentage-based news classifier. Given that LLMs are statistical models, it is hypothesized that they might struggle more with percentage-based classifications, making this an interesting area to investigate:

```
Instruction:
You are a percentage-based news veracity classifier.
Given an article, classify whether it is real or fake
by assigning a value between 0% and 100%,
where 100% means definitely fake and 0% means definitely real. Article:
{ARTICLE}
Classification Percentage Fake:
```

Percentage-based classification gives the most room for nuance in the predictions of LLMs. In combination with the detection strategies above, this can give insight into whether these LLMs can give more nuanced responses than the baselines.

CoT Prompting: The second tier of guidance leverages the Chain of Thought (CoT) prompting strategy [12], which extends beyond simple classification tasks. Instead of merely requesting a classification, this approach prompts the LLM to explicitly reason about the claim or article before providing an answer. This reasoning process has been shown to enhance performance for text classification in prior studies [48]. The methodology aligns with the strategies employed by fact-checkers in FakeHealth and FakeNewsNet.

In our implementation, we adopt three of the four criteria used by PolitiFact (discussed in 2, focusing on the message component of the SMCR model [49]. Due to not having information regarding PolitiFact’s jurisprudence, we were unable to include the fourth criterion related to the source's reputation. The three criteria we apply are:

1. **Literal Truth:** Does the claim align with verifiable facts?
2. **Alternative Interpretations:** Can the statement be understood in different ways?
3. **Supporting Evidence:** Is there evidence provided by the source to substantiate the claim?

The LLM is prompted to evaluate these criteria sequentially, simulating the reasoning process of human fact-checkers. After completing this analysis, the LLM delivers its classification in a binary format, consistent with the As-is Binarcy Classification approach. The prompt we used is the following:

```
Instruction:
You are a binary news veracity classifier.
Given an article you classify whether it is real or fake
answer with "REAL" if it is real and "FAKE" if it is fake.
Before classifying first reason about the 3 following questions regarding the article
and take that reasoning into account when deciding on your classification
Questions:
1. Does the claim align with verifiable facts?
2. Can the statement be understood in different ways?
3. Is there evidence provided by the source to substantiate the claim?
Article:
{ARTICLE}
Classification:
```

Fine-Tuning The final stage of our approach involves fine-tuning the LLMs on the previously mentioned datasets, providing the highest level of guidance [50]. Fine-tuning allows the model's internal parameters to be tailored for optimal performance on the target task, often resulting in substantial improvements. Due to hardware constraints and a desire to reduce our environmental impact, we employ PEFT [19], specifically LoRa fine-tuning, as outlined in prior works. As a comprehensive exploration of the fine-tuning process can warrant its own manuscript, we do not focus on fully optimizing the fine-tuning pipeline here. Instead, we aim to explore the potential of fine-tuning rather than developing the highest-performing model. The prompting format we use for the fine-tuned models is, again, the binary classification method. This is because the ground-truth labels of our combined dataset are binary labels. Therefore, we cannot fine-tune discrete, continuous or CoT prompting techniques.

4.3. Lenses of Exploration

Inspired by research question 3, this study investigates the LLMs from different contexts and perspectives. To give a more complete and contextualized exploration of the role that LLMs can play, we take not just raw performance into account when evaluating the models but also investigate other lenses inspired by earlier works and the SMCR model.

4.3.1. Performance Lens

The first lens we explore is raw performance, which assesses how effectively the models detect or classify fake news. The primary objective of this evaluation is to measure the efficacy of LLMs in distinguishing between real and fake news using robust metrics that capture different aspects of performance:

- **Validity of labels:** As LLMs work as next-token predictors, it could be the case that we ask the models to answer only from a subset of options, and they still respond with another token. This makes their prediction unusable. For this, we introduce the metric of *Validity*. It is measured by the number of valid predictions divided by the total number of predictions.
- **Accuracy:** This crucial metric quantifies the proportion of correct classifications the LLM makes.
- **False Positive Rate (FPR):** This metric measures the frequency at which the LLM incorrectly flags real news as fake. A high FPR can undermine user trust and suppress legitimate news.
- **False Negative Rate (FNR):** Conversely, the FNR reflects the rate at which the LLM fails to detect fake news, potentially allowing misinformation to spread.

These metrics offer a clear understanding of the overall performance of the models across various detection strategies, making it easy to compare them with baseline models. We assess all model predictions using different detection methods, providing a comprehensive way to evaluate differences not only between models but also between different model types i.e. the LLMs and baselines. This approach also yields results that can be easily compared in future research.

4.3.2. Textual Characteristics Lens

The Textual Characteristics lens is inspired by the message related factor of presentation factors described in Chapter 2 and the work of Horne et al. [47], where they use different readability features to train their ML classifiers. Specifically, this lens investigates how article length and the readability of the text impact the models' classifications. We measured the text length of each article to determine if and how this factor influences the LLM's ability to detect fake news. Additionally, we used the Dale Chall score to measure the readability of the news content to get a picture of the influence of textual characteristics on the LLM's classification of Fake News. One possible hypothesis could be that fake news is often short and without much explanation, whereas longer text could be more substantiated and professionally edited. We calculate a variety of widely used readability tests for each article and their text length. We then put these features in different bins and investigate where models behave well, where they make many mistakes and how their predictions align with ground truth and the underlying characteristics of the datasets. This analysis helps to uncover potential limitations related to the text complexity or verbosity of news content.

4.3.3. Psychological Lens

Another lens worthy of investigation is the psychological lens. The message-related factor of psychological factors mainly inspires this lens. More concretely, we analyse whether the sentiment of an article impacts the behaviour and predictions of the LLMs. This gives us insight into the propensity of the models to give less or more sentimental text more often a certain classification. We first classify each article as having a certain sentiment by using the well-established sentiment analysis library *Valance Aware Dictionary and sEntiment Reasoner* (Vader) [51]. Vader is a lexicon and rule-based sentiment analysis tool initially created for analysis on SMNs. We follow the work of Shrestha et al. [31], where they used it for their RF classifier. We first labelled each entry in the dataset as being either *Positive*, *Negative* or *Neutral*. We then evaluated the predictions from the models with different detection strategies. First, we investigate whether there was a notable difference in behaviour on these labels and whether there was a difference in the models' performance on either sentimental or neutral text. Sentimental is defined as text that is either positive or negative. A logical hypothesis would be that more objective news is more often true and that positive or negative news is more often fake with the intent to play on the reader's emotions. We split the predictions into different sentiment labels and investigated how the models behaved correctly and incorrectly and where there was no notable difference. This analysis gives us an overview of how psychological cues can impact the LLMs' predictions.

4.3.4. Domain Lens

As described in Chapter 1 and 2, different domains and topics of articles can have different impacts on the parties mentioned and involved. For celebrities, fake news might harm their reputation or get them ostracized in the public eye. Fake news about vaccines can stop people from getting their vaccinations and possibly impact the herd immunity of a country or community, and misinformation presented as political news can sway elections. Just as the impact on entities affected can differ between context and topic, LLM predictions could be affected by topics. One could hypothesize that tech companies aligned with a certain ideology would train their LLMs to more vigilantly disagree with fake news against that ideology. With the domain lens, we want to investigate possible differences in the behaviour of the models over different contexts and understand the possible impact they could have on automated fake news detection. We split the predictions into different topics and analysed their characteristics from this perspective. We categorized articles by their source: *Entertainment* for GossipCop, *Political* for PolitiFact, and *Health* for FakeHealth and HealthRelease. Because the MOCHEG dataset did not show a clear domain for their articles, we labelled them as the *Undefined* domain. We also do some basic entity analysis to see whether we can find certain topics or words where models behave differently.

4.3.5. Sustainability Lens

Finally, we consider the sustainability perspective, acknowledging that AI and ML models can have significant environmental impacts. We compare the amount of energy consumed for the different models and contextualize this with earlier models to gain insights into the possible cost for the environment of employing. We assessed the carbon emissions and energy consumption associated with different models and detection strategies. We do this by recording the energy used for model inference (querying the models for predictions) and fine-tuning the models. By evaluating the energy consumed compared to the model size, detection strategy applied and fine-tuning, we can gain insights into how to sustainably leverage LLMs for fake news detections. This analysis provides insight into how performance differences translate into environmental costs, offering a critical view of the broader implications of deploying LLMs for fake news detection.

5 Setup

This section describes the concrete process we used to run our experiments and data analysis of our combined dataset labelled fake and real News.

5.1. Process

The process we used for our experiments is closely related to our main methodology described in Chapter 4 but elaborates more on the concrete steps taken, depicted in figure 5.1. We first started with data gathering. We downloaded the FakeNewsNet [44] dataset by following their *.Readme* file in their Github repository[1]. Upon manual examination, we discovered that since the initial creation of FakeNewsNet, some articles were no longer accessible or had been replaced by spam content and advertisements. Through manual investigation, we found that this unusable content was generally repetitive. Therefore, we removed all duplicate and irrelevant articles from the dataset, resulting in a total of 14,398 unique labelled articles. The script provided by Shu et al. had already normalized the ground truth labels to either fake or real, resolving the discrepancies between GossipCop's and PolitiFact's rating systems.

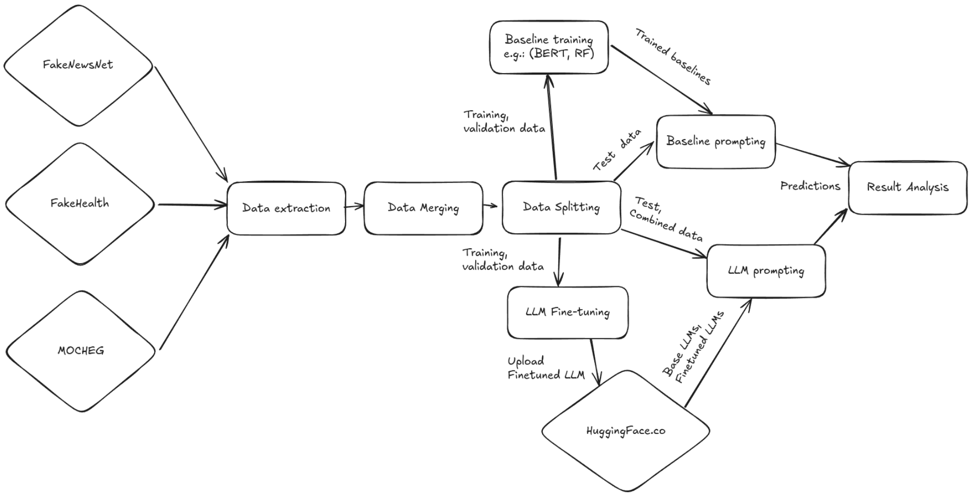


Figure 5.1: Our experiment pipeline

Next, we collected the FakeHealth [35] from their Github repository[2]. As their rating system for the veracity of the articles was on a scale from 1 to 5, we opted to convert these ratings to true when they had a rating of 3 or above. Finally, we downloaded the articles from the MOCHEG [45] dataset from their Github repository[3]. Because MOCHEG also used articles from PolitiFact, we removed all articles from

[1] https://github.com/KaiDMML/FakeNewsNet

[2] https://github.com/EnyanDai/FakeHealth/tree/master

[3] https://github.com/VT-NLP/Mocheg

MOCHEG that were already occurring in FakeNewsNet to remove duplicate articles. We then added topics for every dataset: content from PolitiFact was assigned the *Political* label, content from GossipCop got the *Entertainment* label, FakeHealth was labelled as *Health* and finally since the MOCHEG dataset lacked a clear categorization, we labeled it as *Undefined*.

In the next step, we merged the earlier datasets into one and calculated the length of the articles for each entry. Subsequently, we calculated the sentiment by using the Python Vader sentiment library [51] to mark the articles as either *Negative*, *Positive* or *Neutral*; for this, we took a similar approach as Shrestha et al. [31] used for feature extraction mentioned in Chapter 3. We also followed the implementation of Shrestha et al. to compute multiple readability scores using the Python readcalc library.

We then adjusted the code shared by Shrestha et al., accompanied by their paper [31], to set up and train the RF classifier. Following that, we took inspiration from the work of Wu et al. [37]. In which they leveraged a BERT classifier for their prompting step to set up their own BERT-based model. To train these models for fake news detection, we performed a 60/20/20 split on the dataset from above into a training, validation and test set, respectively. To keep the distribution of characteristics the same as the initial dataset, we performed a stratified split, which is further discussed in the next section. These baselines were then ready to leverage for classification.

Next, we leveraged the six models in combination with the detection strategies described in chapter 4 to classify each news article from the merged dataset mentioned earlier. All LLMs we used were sourced from Huggingface.co[4] and are publicly available to use. Specifically, this means that for each LLM and each detection method, we queried them for their predictions. For the baselines, we only used the binary classification strategy as earlier works for this task mostly use binary classification.

To fine-tune the models, we combined the Binary classification prompt described in chapter 4 with the ground-truth label and used LoRa finetuning. We uploaded our fine-tuned models to huggingface to ease future reproducibility studies and contribute to open science.

Finally, we analyzed the results through multiple lenses with standardized scripts. These results are discussed further in Chapter 6.

5.2. Ethical considerations

In this work we set out to do good scientific research. This means that our work should be able to be validated, reproduced and easily adaptable for future works. Fake news and fact-checking is a topic that has a great impact on people, businesses and society as a whole. Therefore it is especially important that the scientific work done around this, is ethically responsible. In this section, we describe how we direct our efforts to accomplish this.

In terms of data management, we made sure that our combined, training, validation and test sets are all available and stored at a TU Delft-managed drive. Which can be requested by contacting TU Delft. Additionally, we shared our full code on GitLab[5], with a complementary .ReadMe file to explain how to use the pipeline. We also added instructions on how to use and extend our code for future research. Where other researchers can add and extend datasets and models. Lastly, we shared our fine-tuned models on huggingface.co[6].

As mentioned before, fake news can have a serious impact on many parties. Therefore we want to make clear that any results and insights found from our results are generated from our specific methodology, dataset and models used. Insights and conclusions should not be generalized about fake news as a whole but only as an informed starting point for future research. When using LLMs in real life for fake news detection, one should consider that when done wrong, can have serious consequences and can lead to people's suffering.

5.3. Data characteristics

To contextualize the results of our experiments, we describe some of the characteristics of our final dataset and compare it with our test split dataset, which was used to evaluate the fine-tuned models and baselines.

[4] https://huggingface.co

[5] https://github.com/Merlijnmacgillavry/FakeNewsDetectionUsingLLMs

[6] https://huggingface.co/collections/Lord-Papillon/finetuned-fake-news-detection-datasets-66f47b75e799ce793f800d3f

Slice	#Real	#Fake	%Real	%Fake	#Total	%Total
Total	21554	10523	67.19%	32.81%	32077	100.0%
Domains						
Health	1429	726	66.31%	33.69%	2155	6.72%
Entertainment	10144	3648	73.55%	26.45%	13792	43.0%
Politics	311	295	51.32%	48.68%	606	1.89%
Undefined	9670	5854	62.29%	37.71%	15524	48.4%
Sentiments						
Positive	12322	4885	71.61%	28.39%	17207	53.64%
Negative	5352	3254	62.19%	37.81%	8606	26.83%
Neutral	3880	2384	61.94%	38.06%	6264	19.53%

Table 5.1: Veracity characteristics of the final dataset with different domain and sentiment splits.

Table 5.1, shows the characteristics of our final dataset. We denote the number of news articles labelled as either real or fake and their proportions. Furthermore, we divide the dataset into slices of domains to contextualize the domain lens and sentiments to contextualize the psychological lens. The complete dataset has around two times more real articles than fake ones. Although skewed, this is either a comparable distribution or a more balanced distribution than the earlier works on automated fake news detection discussed in 3.

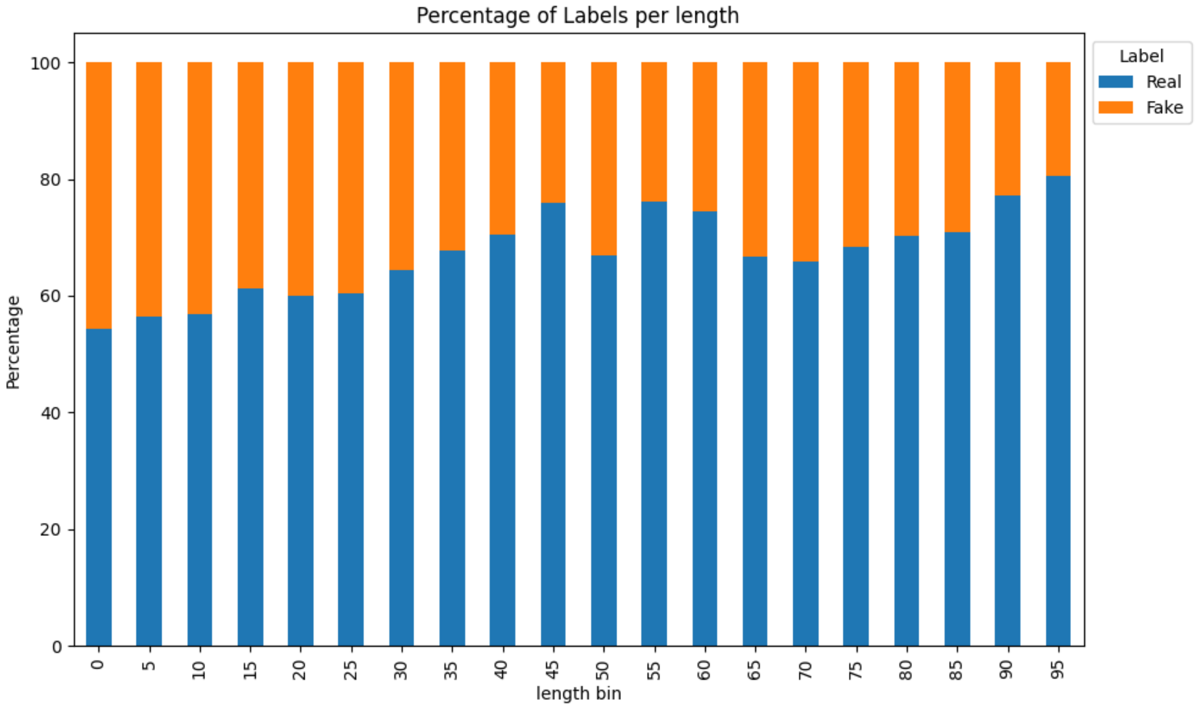
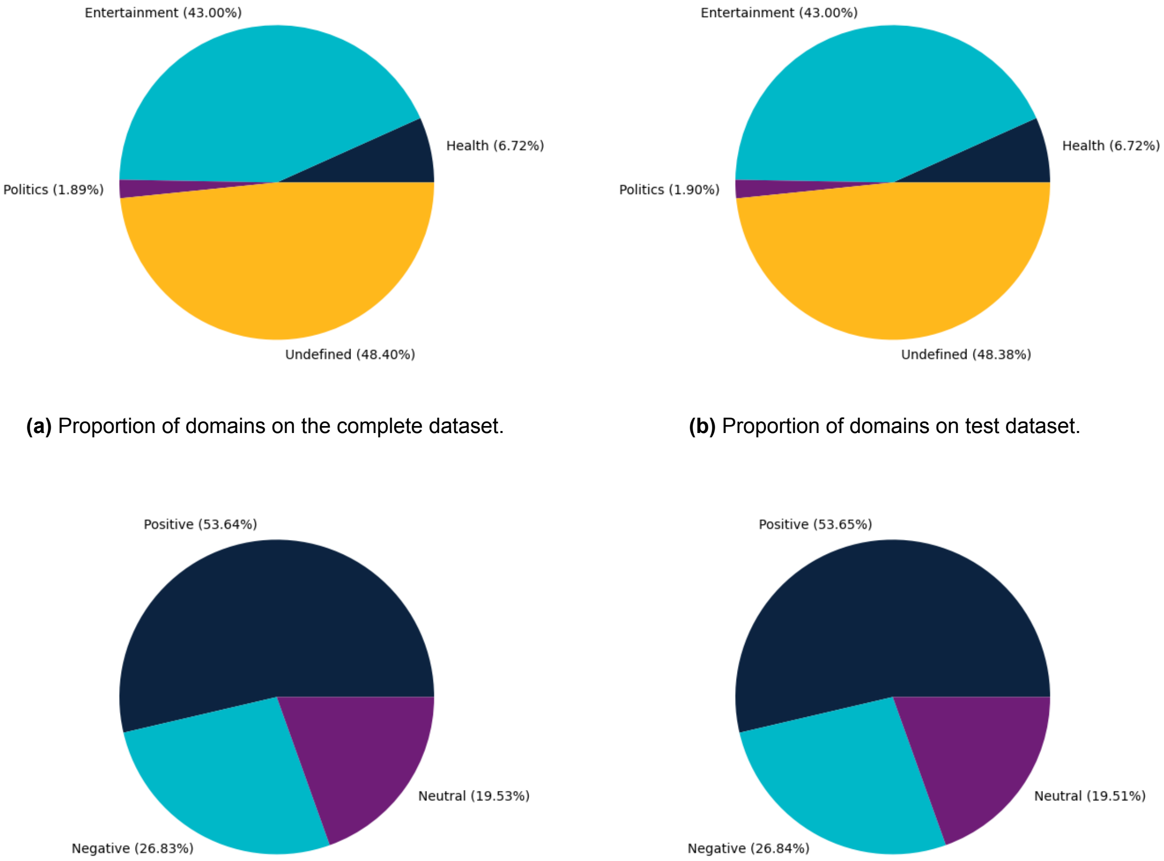


Figure 5.2: News distributions of veracity over text length bins of complete dataset

In terms of domains, the entertainment and undefined news articles are the most common, with them combined, representing half of the dataset. Political news, on the other hand, is the least common type evaluated. However, the true proportion of this topic is likely bigger, as the MOCHEG dataset, which we labelled as undefined, includes news from both PolitiFact and Snopes, both of which fact-check political news. Despite this, accurately assigning the correct domain would require a thorough analysis, which is beyond the scope of this work and could serve as the focus of a separate study. The proportion of real and fake news articles for the different domain slices is comparable to that of the total set, with the entertainment domain having more real articles and the politics dataset having a more equal distribution.

The proportions for the different sentiments are also imbalanced. A little bit more than half are positive, a quarter are negative, and the neutral articles represent around a fifth of the total news articles. For the positive articles most of them are real with a proportion of 70% and around 30% fake. For both negative and neutral the veracity labels are more balanced, with around 60% labelled as real and 40% fake.

We also investigated the distribution of real and fake news over the text length of the articles. We did this compare with results from our textual characteristics lens. Figure 5.2 depicts the twenty 5% bins distribution of the articles in terms of text length and veracity proportions. We can see that there is a small tendency for the news to be more often real as the text length increases.



(a) Proportion of domains on the complete dataset.

(b) Proportion of domains on test dataset.

(c) Proportion of sentiments on the complete dataset.

(d) Proportion of sentiments on test dataset.

Figure 5.3: Distributions of sentiments and domains for complete and test dataset.

As described in the previous section we applied a stratified split on the complete dataset to create training, validation, and test sets for fine-tuning and training of the models. Because we want to ensure that our test dataset on which the fine-tuned models are evaluated has the same characteristics as our complete dataset, we compared the domain and sentiment distributions for both models in Figure 5.3. As shown both datasets have almost identical distributions. This equal distribution should lead to the models reaching similar results if the models are consistent in their classification. If we see big differences in performance, we can deduce that the models are, in fact, not classifying news articles consistently.

Part IV

Results & Discussion

6 Results

In this chapter, we report the results of our experiments. We discuss the models' results and behaviours through 5 different lenses. With these lenses we aim to give a more holistic view of the results. We start with Overall Performance, followed by the Textual Characteristics lens, Psychological lens, Domain lens and finally the sustainability lens.

6.1. Overall Performance

The overall performance is shown in Table 6.1. Each row denotes a model with a certain detection strategy. The *Model* column describes the model and the detection strategy we use for evaluation. The *Size* denotes the number of parameters; Gemma-2b-it, for example, has a parameter size of around 2 billion.

The table also describes the number of rows that models predicted for each entry. The number of rows (articles evaluated on) is 32077 or 6416, referring to the combined dataset and the test set split, respectively. Because the baselines and fine-tuned models must be trained and validated, we can only evaluate them on the test set. Otherwise, we evaluate them based on data on which they have already been trained. There is only one detection strategy for the baseline models: binary classification. Because the different LLMs are trained on next-token prediction, they are not guaranteed to respond with the labels that we expect. For example, we ask the LLMs to only respond with real or fake in binary classification. When they do not, we label their prediction as invalid. The *Valid* column denotes the fraction of valid predictions generated by the models. We also keep track of the accuracy and accuracy of valid predictions as *Acc.* and *Acc. V* respectively.

We also denote the fraction of false positives and false negatives for each row to investigate how the models misclassify news articles. Finally, to compare the distribution of the predictions of the various models, we also denote the fraction of predictions the model makes as either fake or real.

Each row shows a model combined with a detection strategy. Some detection strategies, namely percentage and discrete prompting, are converted to binary classification labels to calculate their accuracy. Therefore, we take different cut-off values. We use two cut-off values for the discrete detection method: *Includes True* and *Above Mostly True*. *Includes true* means that whenever a classification includes the word "true", we label the prediction as real. *Above mostly true* means that the prediction must be either "mostly true" or "true" to be labelled as real. As with the discrete detection strategy, we also have two cut-off values for the continuous detection strategy: *Above 50* and *Above 75*, marking every prediction as fake when above the corresponding percentage.

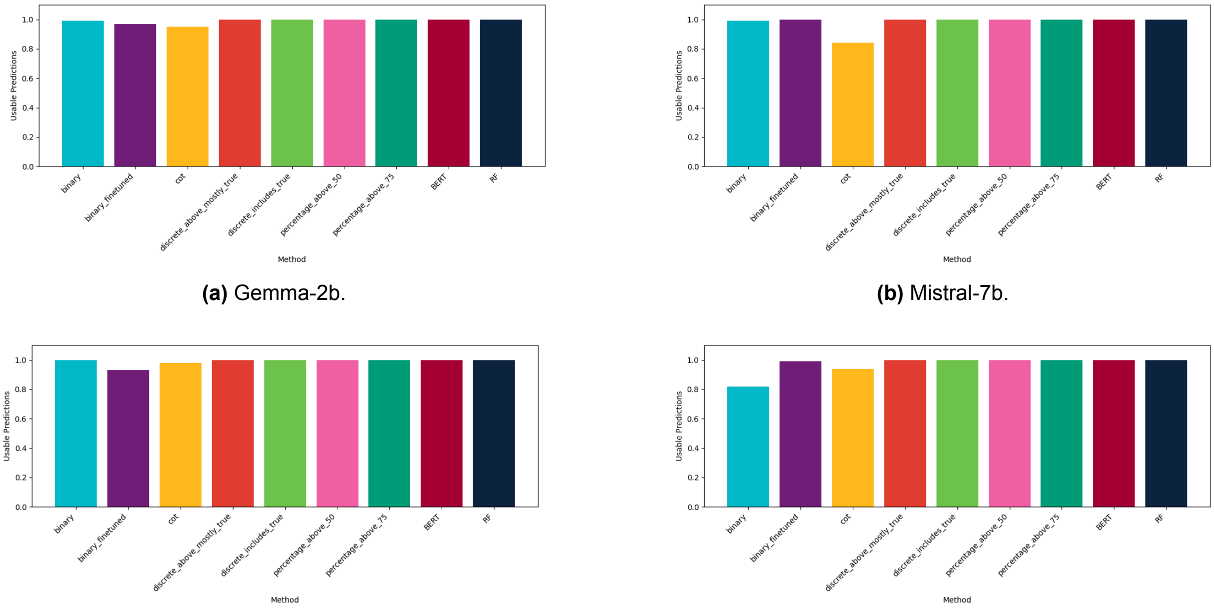
In general, the baselines perform relatively well compared to the LLMs. With a perfect fraction of valid labels and an accuracy higher than many LLMs.

Model	Size	#Rows	Valid	Acc.	Acc. V	FP	FN	%Fake	%Real
Baselines									
BERT	N/A	6416	1	0.77	0.77	0.11	0.33	0.33	0.67
RF	N/A	6416	1	0.71	0.71	0.04	0.25	0.11	0.89
LLMS									
Gemma-2b-it									
Binary	2b	32077	0.99	0.55	0.56	0.33	0.12	0.54	0.46
Binary	2b	6416	0.99	0.57	0.58	0.29	0.13	0.49	0.51
Fine-tuned	2b	6416	0.97	**0.65**	**0.67**	0.18	0.15	0.36	0.64
CoT	2b	32077	0.95	0.44	0.46	0.47	0.06	0.74	0.26
Discrete Mostly True	2b	32077	**1.0**	0.47	0.47	0.46	**0.08**	0.32	0.68
Discrete True	2b	32077	**1.0**	0.63	0.63	**0.17**	0.2	0.3	0.7
Percentage 50	2b	6416	**1.0**	0.51	0.51	0.3	0.19	0.44	0.56
Percentage 75	2b	6416	**1.0**	0.52	0.52	0.3	0.18	0.44	0.56
Mistral-0.2-7b-it									
Binary	7b	32077	0.99	0.72	0.72	0.07	0.2	0.2	0.8
Binary	7b	6416	0.99	0.72	0.72	0.07	0.21	0.19	0.81
Fine-tuned	7b	6416	**1.0**	**0.81**	**0.81**	**0.06**	0.13	0.26	0.74
CoT	7b	32077	0.84	0.63	0.75	0.03	0.23	0.1	0.9
Discrete Mostly True	7b	32077	**1.0**	0.54	0.54	0.38	**0.08**	0.26	0.74
Discrete True	7b	32077	**1.0**	0.61	0.61	0.3	0.09	0.63	0.37
Percentage 50	7b	32077	**1.0**	0.28	0.28	0.54	0.18	0.87	0.13
Percentage 75	7b	32077	**1.0**	0.28	0.28	0.53	0.19	0.86	0.14
Llama-3.1-8b-it									
Binary	8b	32077	**1.0**	0.7	0.7	**0.01**	0.29	0.04	0.95
Binary	8b	6416	0.99	0.7	0.7	**0.01**	0.29	0.04	0.95
Fine-tuned	8b	6416	0.93	0.7	**0.75**	0.02	0.23	0.07	0.91
CoT	8b	6416	0.98	0.67	0.68	0.0	0.32	0.01	0.99
Discrete Mostly True	8b	32077	**1.0**	0.55	0.55	0.37	**0.08**	0.56	0.44
Discrete True	8b	32077	**1.0**	0.66	0.66	0.2	0.14	0.39	0.61
Percentage 50	8b	32077	**1.0**	0.46	0.46	0.29	0.24	0.38	0.62
Percentage 75	8b	32077	**1.0**	0.47	0.47	0.29	0.25	0.37	0.63
Gemma-2-9b-it									
Binary	9b	32077	0.74	0.48	0.64	0.29	0.06	0.58	0.42
Binary	9b	6416	0.82	0.54	0.66	0.27	0.07	0.53	0.47
Fine-tuned	9b	6416	0.99	**0.8**	**0.81**	**0.04**	0.15	0.21	0.79
CoT	9b	6416	0.94	0.6	0.63	0.27	0.1	0.5	0.5
Discrete Mostly True	9b	32077	**1.0**	0.39	0.39	0.59	**0.02**	0.7	0.2
Discrete True	9b	32077	**1.0**	0.45	0.45	0.51	0.04	0.8	0.2
Percentage 50	9b	32077	**1.0**	0.6	0.6	0.3	0.1	0.46	0.54
Percentage 75	9b	32077	**1.0**	0.61	0.61	0.23	0.16	0.33	0.67

Table 6.1: Overall Performance of LLMs for Fake News Detection

Validity Of Labels The validity of labels is an important characteristic when evaluating models' performance. A higher validity means that a model more often gives a classification that can be easily programmatically extracted. Depicted in Figure 6.1, it is evident that not all LLMs yield the same number of valid predictions over different detection methods. For instance, the Gemma-2b model generates a high proportion of valid labels, with the fraction of valid predictions exceeding 97% for all predictions. Interestingly, the fine-tuned version of the Gemma-2b model has a lower fraction of valid classifications than the base model despite being specifically trained to provide usable answers. This phenomenon occurs with Gemma-2b and Llama-3.1, whereas the other LLMs typically show increased usability when fine-tuned. Mistral only has a relatively low fraction of valid labels for the CoT detection strategy.

Compared to its older and smaller counterpart in the same family, Gemma-2-9b demonstrates significantly lower usability in the binary classification method, achieving the lowest number of valid labels in the entire table at 74%. Among all LLMs, the Mistral model exhibits the highest fraction of valid predictions across all detection methods, with no strategy falling below 99% except for the CoT strategy.



(a) Gemma-2b.

(b) Mistral-7b.

(c) Llama-3.1-8b.

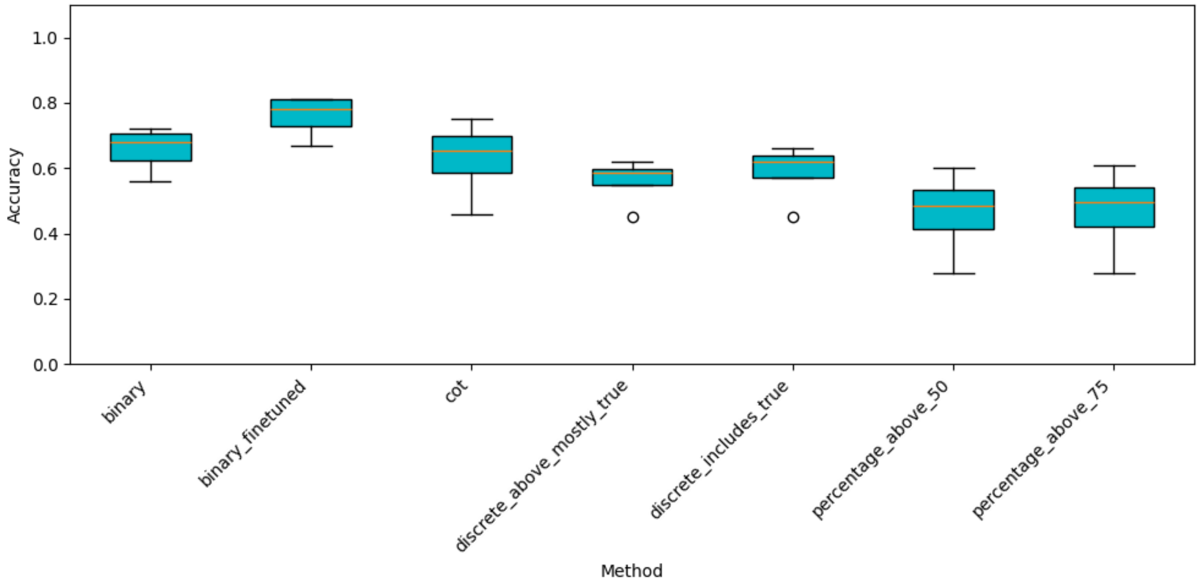
(d) Gemma-2-9b.

Figure 6.1: Fraction of valid labels for LLMs models over all detection methods, compared to baselines.

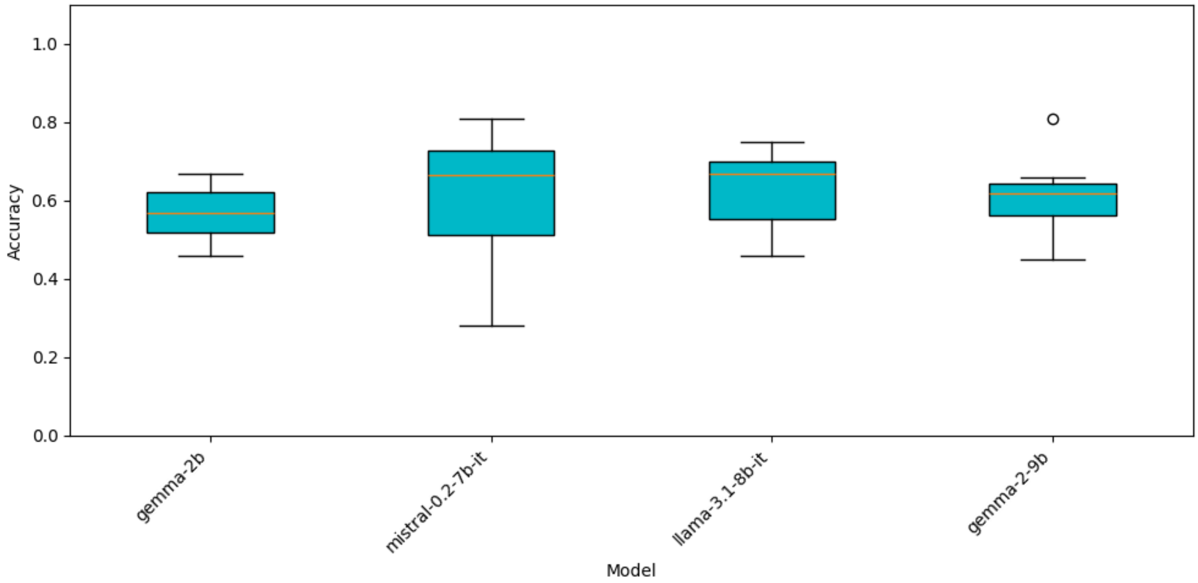
Accuracy Visualized in Figure 6.3, the baselines demonstrate relatively good performance in terms of raw accuracy. The RF baseline, with an accuracy of 71% and the BERT baseline even higher with 77%, outperforms some LLMs in the binary detection strategy and also exceeds their performance when using discrete and percentage-based methods. In the case of Gemma-2b, the CoT detection method produces the least accurate predictions. However, the discrete method with a ”includes true” cut-off outperforms all other detection methods except for the fine-tuned model. The percentage-based detection methods achieve an accuracy of around 50%, which is not particularly impressive.

When examining Mistral, we find that different detection methods yield better results. For example, the binary detection methods outperform both the discrete and percentage-based strategies. Notably, the percentage-based detection methods produce the lowest accuracy predictions in the entire table, with only 28% being correct. Among all the LLMs, Mistral scores highest in terms of accuracy for both the base model in binary classification and its fine-tuned version.

When looking at the box-plots of avarage accuracy of all the detection methods (depicted in Figure 6.2a, we see that fine-tuned binary classification performs best and the percentage based detection methods perform worst. CoT can in some instances be higher but not with a great margin and the discrete models seem to perform the most average. If we look at the box-plots of the models (Figure 6.2b), we see that different models have greater spread in their accuracy. Mistral, the best performing model also has the greatest spread, whereas the Gemma models have smaller spread but seem to have lower accuracy on average.



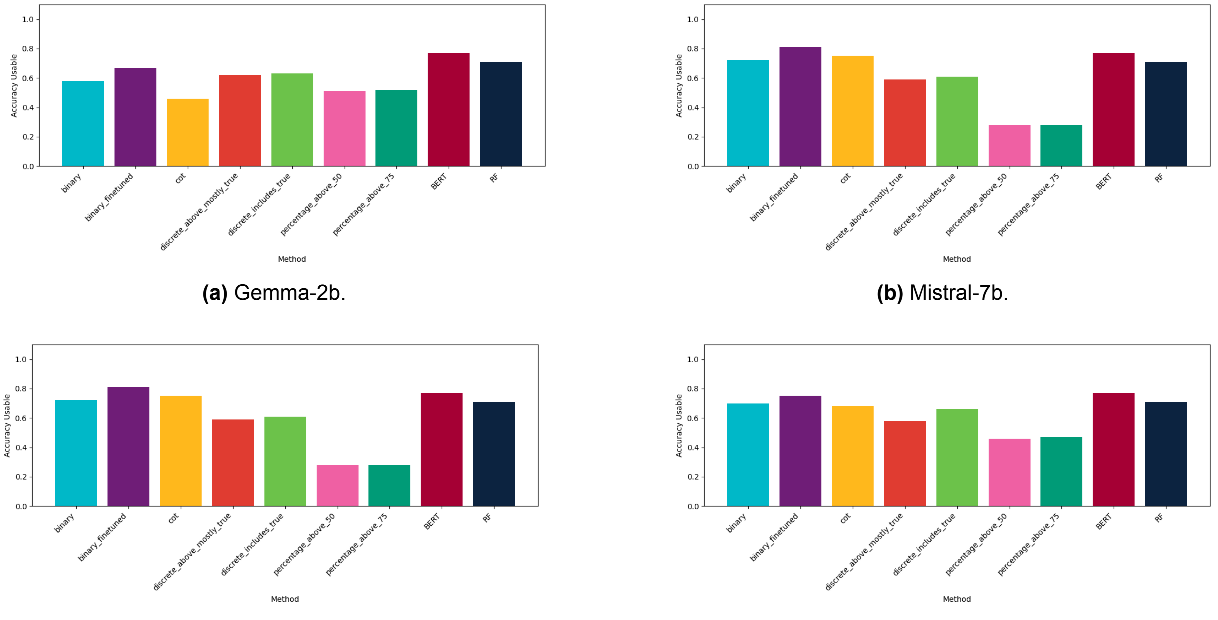
(a) Average accuracy of detection methods.



(b) Average accuracy of LLMs.

Figure 6.2: Average accuracy of models and detection methods.

In contrast, Llama-3.1 performs at an average level, showing no significant outliers compared to the other LLMs. However, Gemma-2-9b exhibits poor binary and discrete classification accuracy with the base model. The smallest and oldest model, Gemma-2b, clearly has the lowest accuracy compared to the other models. Additionally, the base binary classification of Gemma-2-9b is relatively low compared to the Mistral and Llama models.



(a) Gemma-2b.

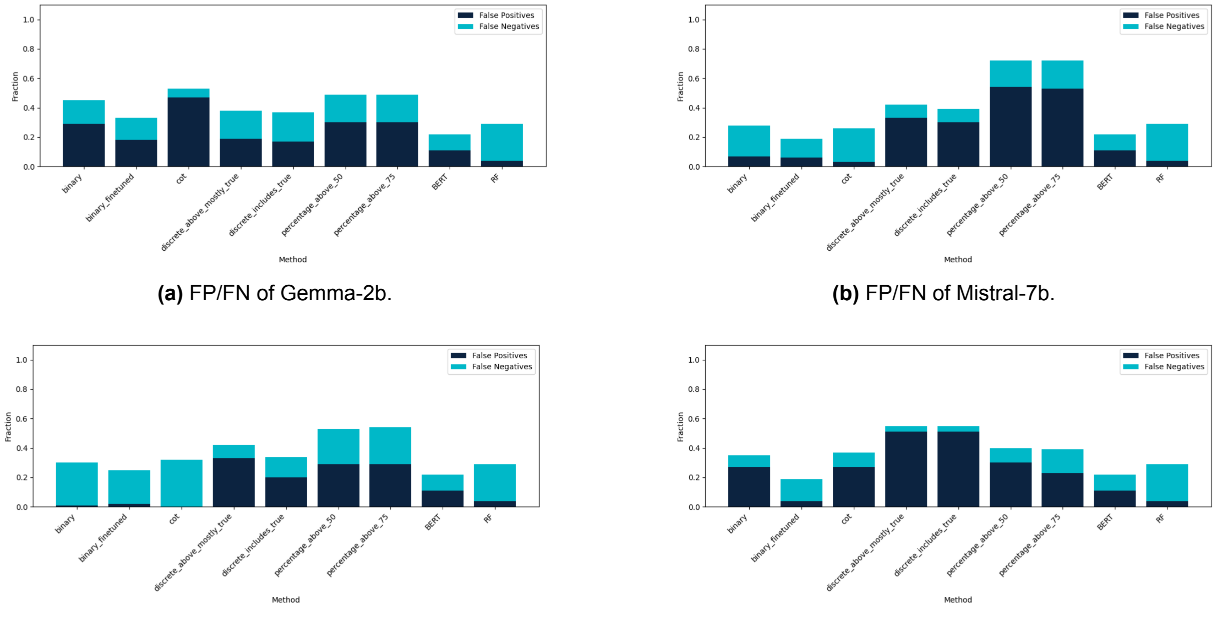
(b) Mistral-7b.

(c) Llama-3.1-8b.

(d) Gemma-2-9b.

Figure 6.3: Accuracy of LLM models over all detection methods, compared to baselines.

False Positives and False Negatives Accuracy tells us a lot about how the different models and detection strategies classify news articles correctly. But to understand how the different models fail to classify news correctly, we need to investigate their false positives and false negatives. In our study, a false negative is an article incorrectly classified as true news, and a false positive is a news article incorrectly classified as fake.



(a) FP/FN of Gemma-2b.

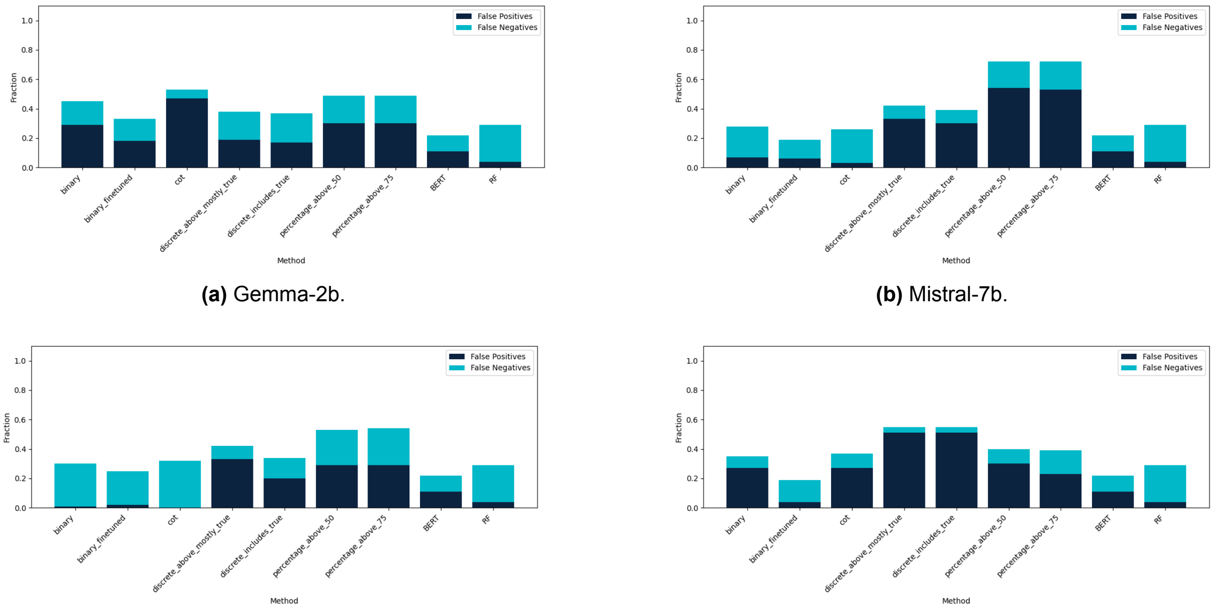
(b) FP/FN of Mistral-7b.

(c) FP/FN of Llama-3.1-8b.

(d) FP/FN of Gemma-2-9b.

Figure 6.4: FP/FN of LLM models.

As Depicted in Figure 6.5, the way different models misclassify news varies. LLama-3.1, for instance, demonstrates very few false positives for both the binary and CoT detection strategies. However, when examining the distribution of predictions between fake and real articles, we observe a tendency toward classifying articles as real. For the discrete and percentage-based techniques, the proportion of articles classified as fake and real is more balanced, though this comes with a notable increase in false positives. The baselines show lower rates of false positives compared to most LLMs but exhibit higher false negatives. The best-performing models Mistral-7b fine-tuned and Gemma-2-9b fine-tuned—achieve very low false positive and false negative rates, while maintaining a more balanced distribution of veracity predictions. Overall, it appears that some models are more prone to false negatives, while others are more susceptible to false positives. Mistral and LLama models tend to favor false negatives, whereas the Gemma models are more likely to produce false positives.



(a) Gemma-2b.

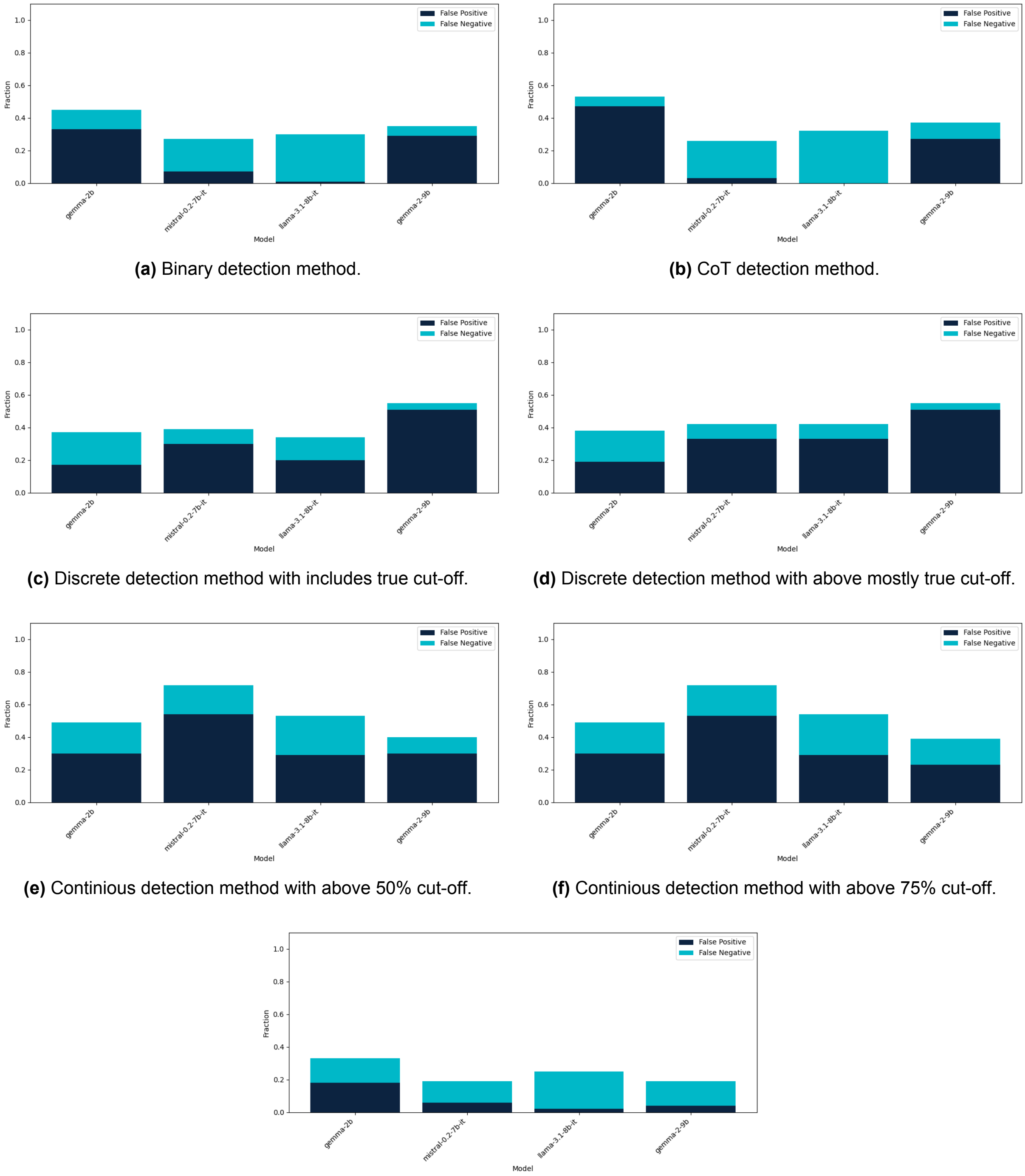
(b) Mistral-7b.

(c) Llama-3.1-8b.

(d) Gemma-2-9b.

Figure 6.5: Misclassifications of LLM models.

Figure 6.6 illustrates the fraction of false positives and false negatives for each detection method across all models. The results show that the CoT and binary detection methods perform worse in detecting fake news as true, whereas the discrete detection methods face more difficulty in identifying true news as fake. When comparing the false positive and false negative rates of the binary detection method between the baseline models and the fine-tuned models, we observe that the most prominent type of false classification is reduced.



(a) Binary detection method.

(b) CoT detection method.

(c) Discrete detection method with includes true cut-off.

(d) Discrete detection method with above mostly true cut-off.

(e) Continious detection method with above 50% cut-off.

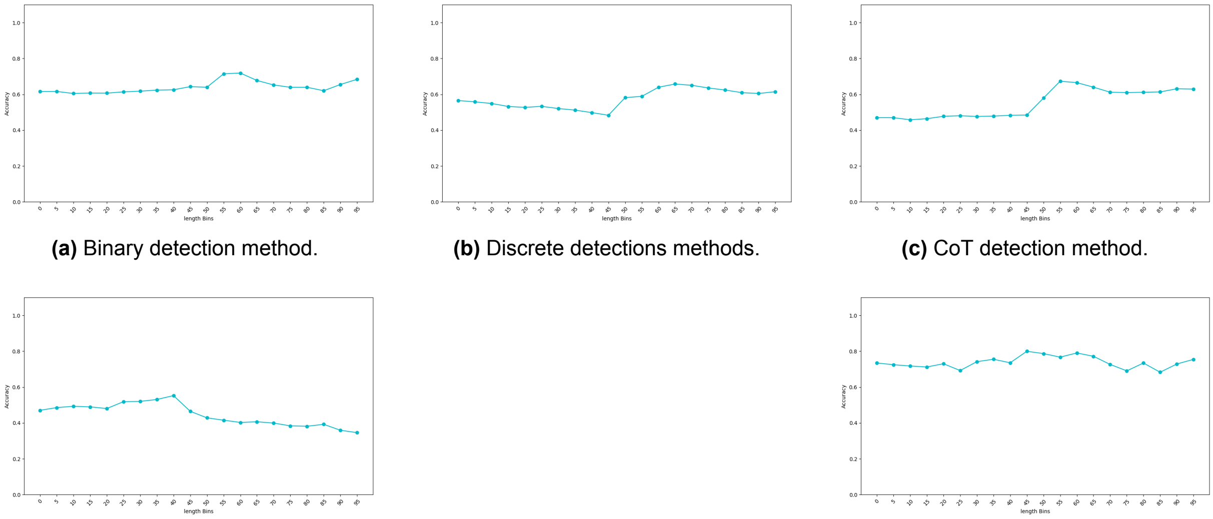
(f) Continious detection method with above 75% cut-off.

(g) Binary detection method with fine-tuned models.

Figure 6.6: Misclassification of detection methods.

6.2. Textual Characteristics

The second lens we investigate is the Textual Characteristics. We do this by analysing two characteristics: text length and readability. Specifically, we want to investigate whether the length of the text and its readability impact the accuracy and false positive/negative rate of the predictions.



(a) Binary detection method.

(b) Discrete detections methods.

(c) CoT detection method.

(d) Continous detection methods.

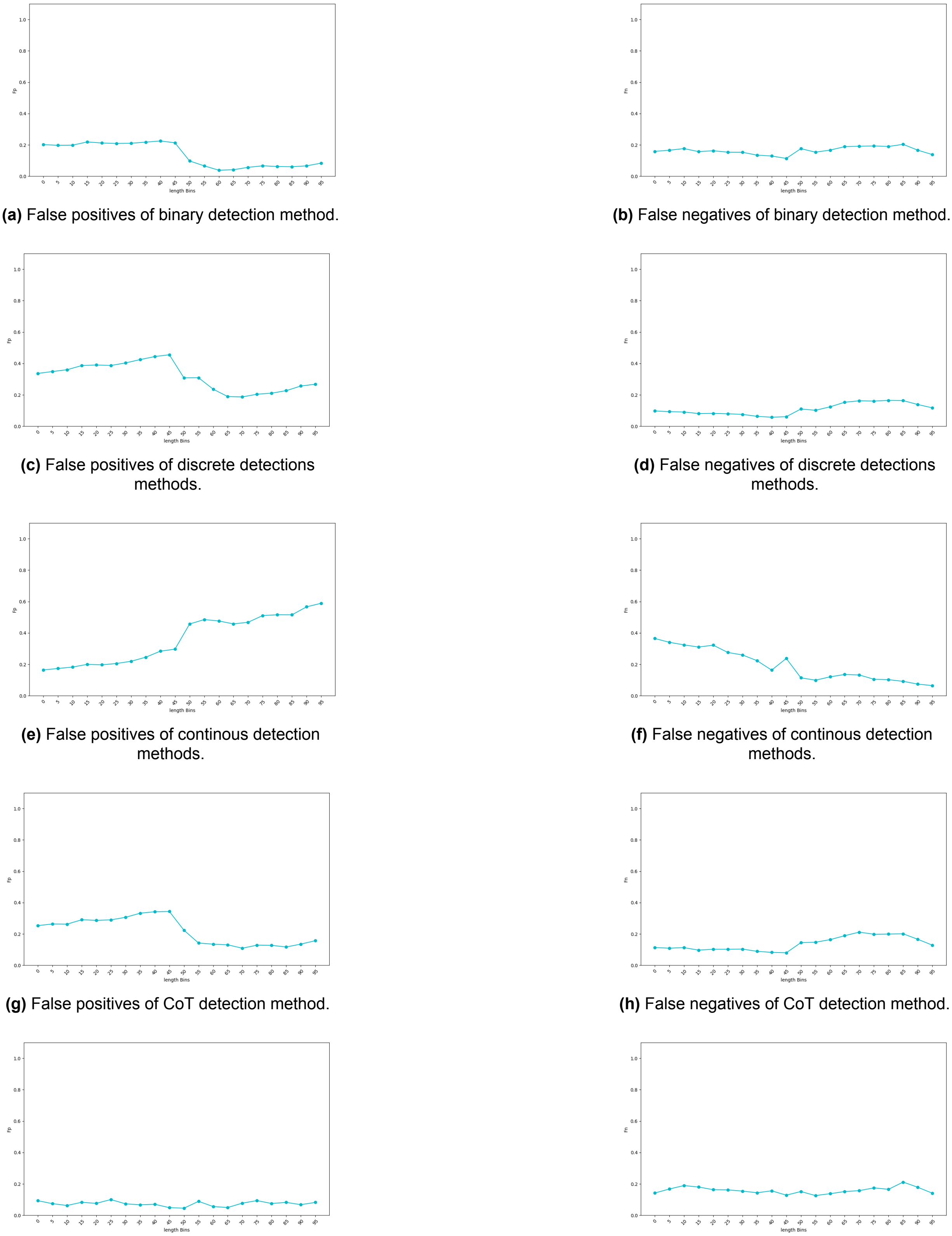
(e) Binary detection method with fine-tuned models.

Figure 6.7: Accuracy of detection strategies along text length portions of 5%.

Text length To evaluate the models' predictions regarding text length, we split the datasets into bins of their percentage ranges. Figure 6.7 and 6.8 show the average valid accuracy and misclassification, respectively, of the different models using different detection strategies based on text length. The x-axis denotes the different percentage ranges of text length. For example, the first bin describes the articles with the 5% lowest text length, and the last bin describes the articles which are longer than 95% of the other articles. The y-axis shows the average valid accuracy or misclassifications for each model. This way, we can compare all the different models and find trends and interesting insights.

First, we look at the accuracy of the different detection strategies regarding text length in Figure 6.7. The binary detection method shows an almost horizontal line with a small gain in accuracy between the 50% and 70% text length mark. After this, it slowly decreases again, with a small uptick for the longest text lengths. We see a similar pattern for the CoT detection method. With the discrete detection methods, we see a small decline till the 45% mark and, afterwards, a rising accuracy. The continuous detection method exhibits a small increase in accuracy till the 40% mark and slowly declines. Interestingly, these patterns seem to flatten for the fine-tuned models. For the different models, regarding average accuracy and text length, we find no notable difference in accuracy with different text lengths.

When we look at the average fractions of misclassifications of the different detection strategies, depicted in Figure 6.8, we see something interesting. For the binary, CoT and discrete strategies, the false positives are higher before the 50% text length mark and lower after, with an obvious drop at the 50% mark. We see the inverse happening after the halfway point for the continuous detection methods. The accuracy slowly increases from the smaller length bins until the 50% mark; it then increases with a jump and then keeps steadily increasing. As with the accuracy, the false positives of the fine-tuned models flatten over the text length. The opposite happens for the false negatives regarding text length compared to the false positives. For most detection methods, we see the false negatives slowly decreasing until the halfway mark, then stepping to a higher fraction and slowly increasing. As with the false positives, the continuous detection method behaves differently and keeps slowly decreasing regarding text length. As with the false positives, the fine-tuned models are again flattened. When investiging the average accuracy of all the models we see similar results as the various for the detection strategies.



(a) False positives of binary detection method.

(b) False negatives of binary detection method.

(c) False positives of discrete detections methods.

(d) False negatives of discrete detections methods.

(e) False positives of continous detection methods.

(f) False negatives of continous detection methods.

(g) False positives of CoT detection method.

(h) False negatives of CoT detection method.

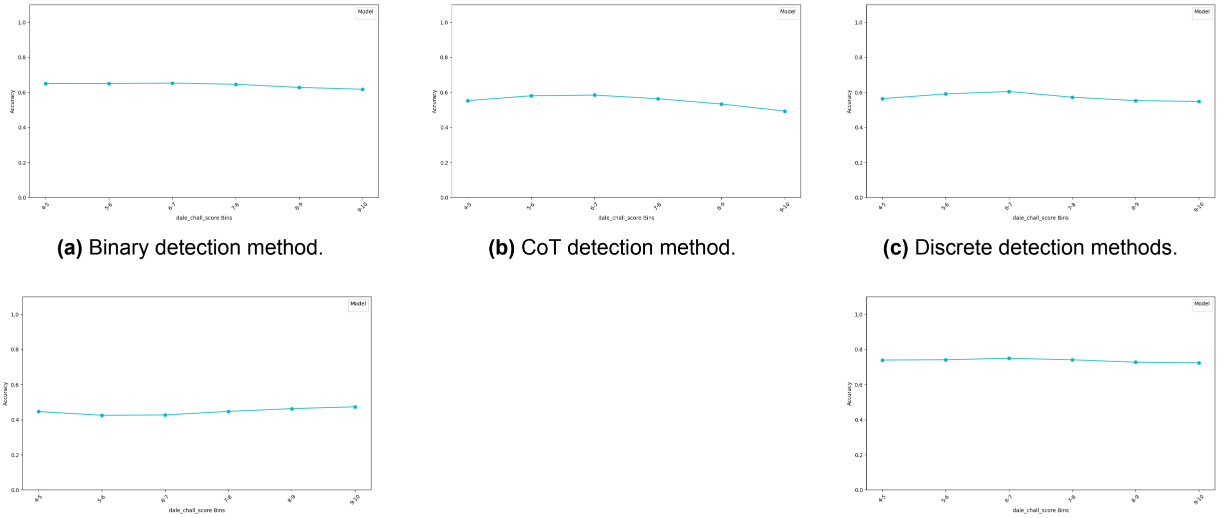
(i) False positives of binary detection method with fine-tuned models.

(j) False negatives of binary detection method with fine-tuned models.

Figure 6.8: Misclassifications of detection methods along text length portions of 5%.

Readability Readability is a characteristic of text that measures how easy it is to understand a given text. Many metrics calculate readability in various ways and with various scores. For many of them, their score can translate to a grade of the U.S.A. grade system. These include readability scores such as: *Flesch-Kincaid grade level*, *SMOG index*, *Gunning fog index* and *Dale-Chall score*. To calculate the readability scores, we used the Python readcalc library. In our evaluation we use the Dale-Chall score to measure readability. A Dale-Chall score for readability is measured between 4 and 10. A score of 4 means that the text is easily understandable by an average 4th-grade student or lower (ages 9 or 10), and a score of 10 means that the text is understandable by an average college student. A higher Dale-Chall score means a text is generally harder to understand. As with the text lengths, we divided the dataset into bins, but now, each bin represents a Dale-Chall score. This way, we can compare the accuracy and misclassifications of the different detection strategies and models to their readability.

Figure 6.9 depicts the accuracy compared to the readability for the different detection methods. For most models, there exists little difference in accuracy compared to different readability scores. It seems the accuracy using the binary, CoT, discrete methods are slightly decreasing when the readability increases. With the continuous detection methods, we see a small decrease in accuracy around the 5-6 score mark. Overall, no visible trends or patterns stand out when comparing readability with readability.



(a) Binary detection method.

(b) CoT detection method.

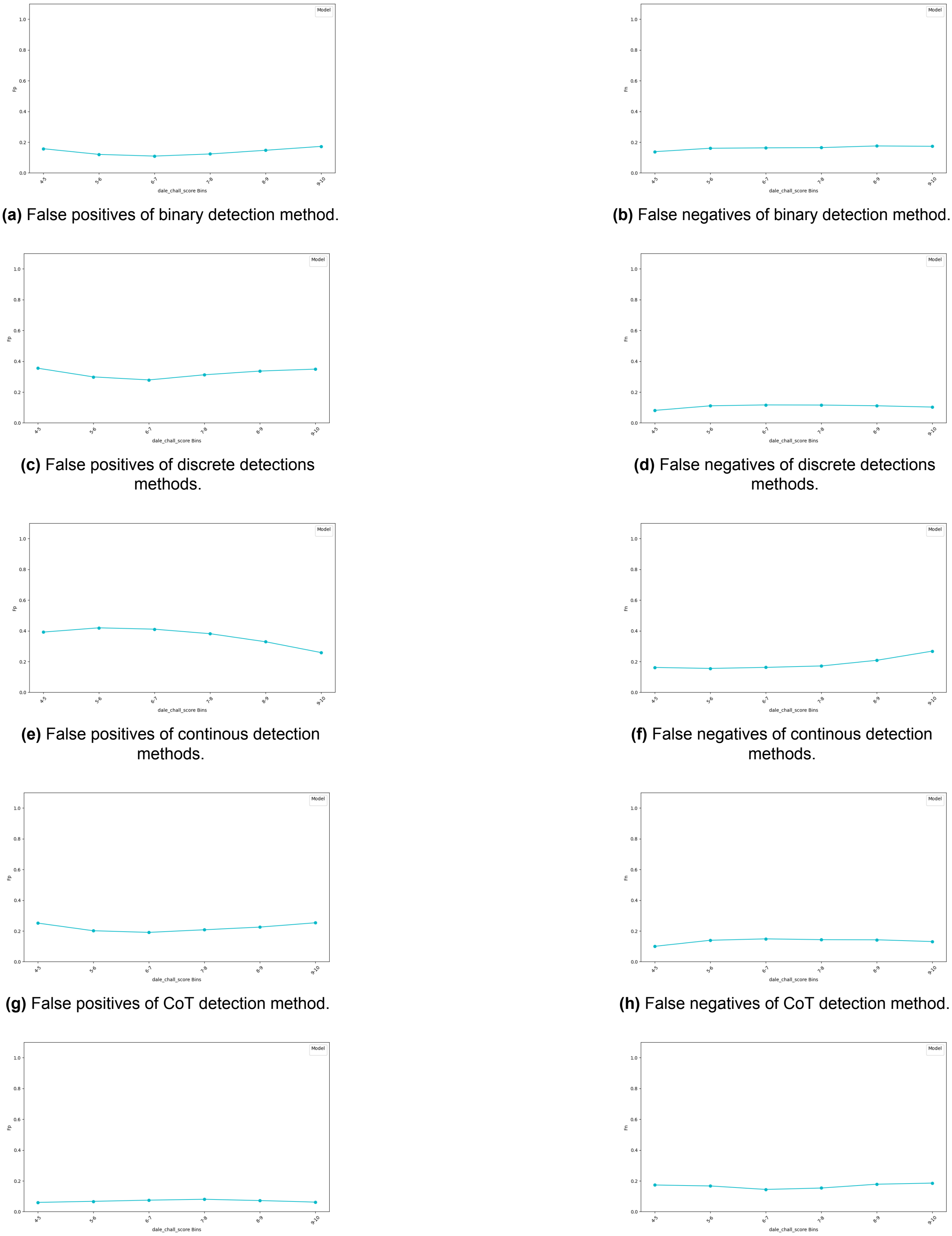
(c) Discrete detection methods.

(d) Continious detection methods.

(e) Binary detection method for fine-tuned models.

Figure 6.9: Accuracy of various detection methods along readability scores measured with Dale-Chall score.

As with the investigation in text length, there are visible patterns when investigating the average misclassifications for different detection strategies dependent on readability, depicted in Figure 6.10. For the binary, CoT, and discrete strategies, the average false positives decrease until a readability score of 7, after which they increase. With the continuous detection methods, we see that the false positives increase till the 6 mark and then drop off as the score increases. For the fine-tuned models, see a small increase of false positives until a score of 8 and then see it decrease again. For the false negatives of the different detection methods, we see that they slightly increase to a score of 6 for the binary, CoT and Discrete detection methods. In contrast, the false negatives of the continuous detection methods increase from a score of 8 onwards. We see the inverse pattern of the false positives for the false negatives of the fine-tuned models.



(a) False positives of binary detection method.

(b) False negatives of binary detection method.

(c) False positives of discrete detections methods.

(d) False negatives of discrete detections methods.

(e) False positives of continous detection methods.

(f) False negatives of continous detection methods.

(g) False positives of CoT detection method.

(h) False negatives of CoT detection method.

(i) False positives of binary detection method with fine-tuned models over text length.

(j) False negatives of binary detection method with fine-tuned models over text length.

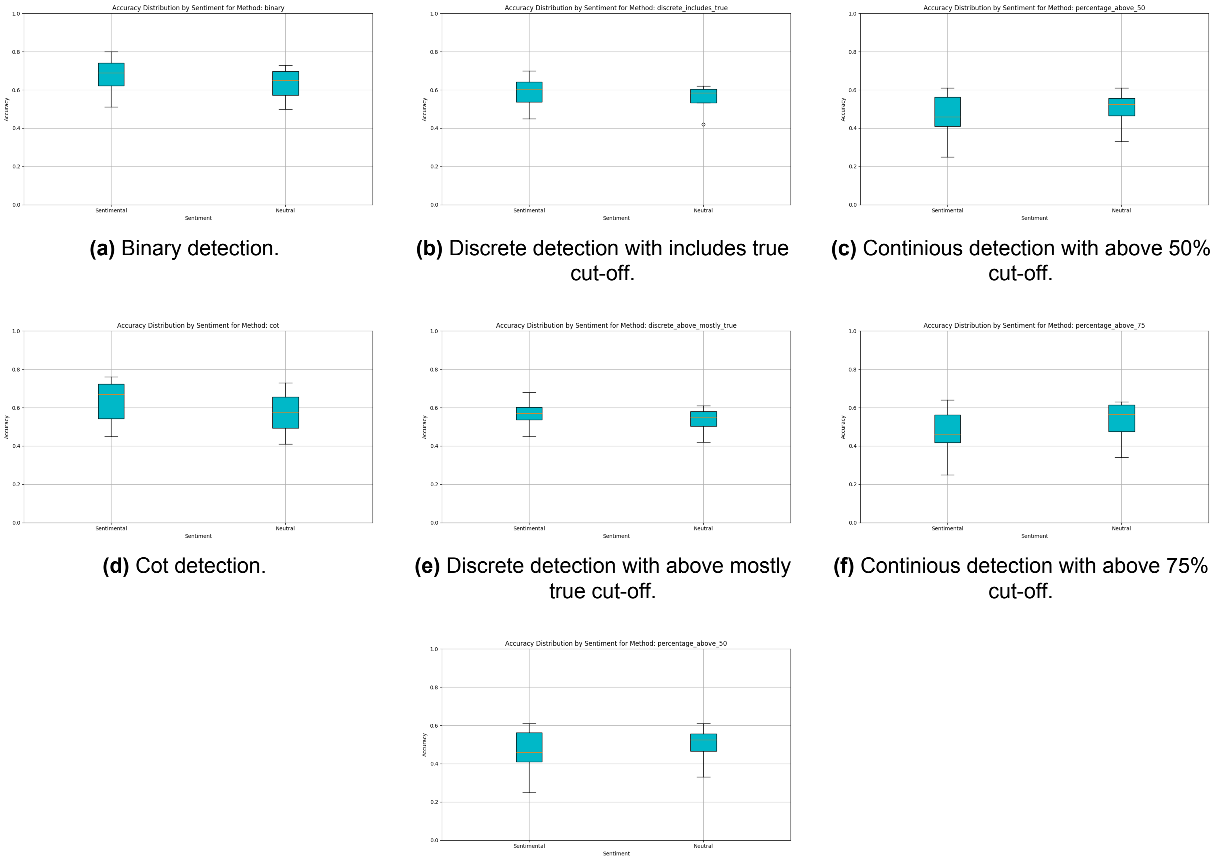
Figure 6.10: Misclassifications of various detection methods along readability scores measured with Dale-Chall score.

6.3. Psychology

In this section, we present the results of our analysis through the psychological lens, focusing on the effect of news sentiment on classification by our models. As described in the methodology, the sentiment of the articles was categorized into three main groups: neutral, negative, and positive. We first examine the classification performance for neutral versus sentimental articles (either positive and negative combined), and then we separately investigate the effects of positive and negative sentiment on classification outcomes.

Neutral Vs Sentimental We first examine how the various detection strategies performed when classifying either sentimental or neutral text (depicted in Figure 6.11). The results indicate that with the binary, CoT, and discrete detection strategies the models encountered greater difficulty in accurately classifying neutral news compared to sentimental text.

On the other hand, the continuous detection strategies present a more complex pattern depending on the cut-off values used. For the cut-off value set above 50%, there is no visible difference in accurate classifications between neutral and sentimental text, indicating that the model maintains consistent accuracy despite sentiment. However, when we apply a cut-off value above 75% we observe that neutral text, is on average, classified better than sentimental text. The fine-tuned models show no visible difference in accurate classification between the two groups but show an improvement in average accuracy.



(a) Binary detection.

(b) Discrete detection with includes true cut-off.

(c) Continious detection with above 50% cut-off.

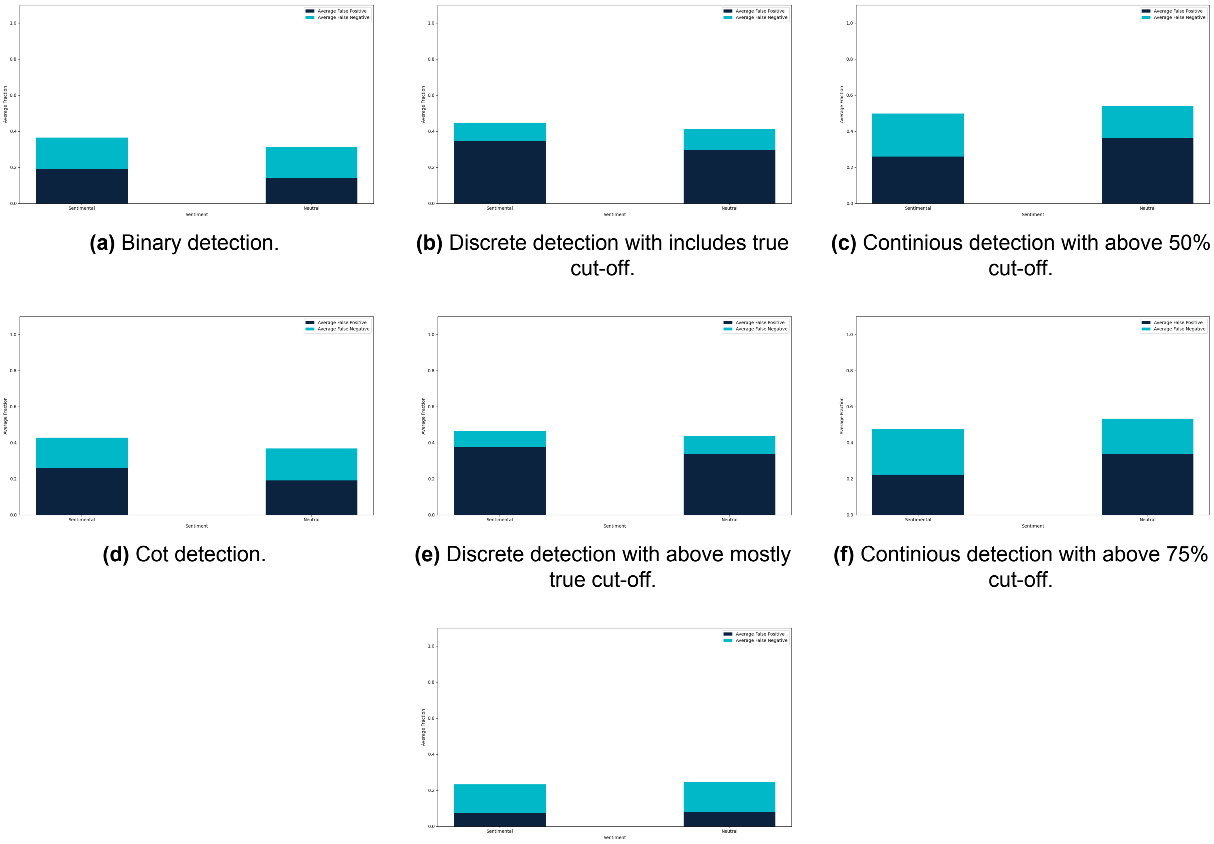
(d) Cot detection.

(e) Discrete detection with above mostly true cut-off.

(f) Continious detection with above 75% cut-off.

(g) Binary detection with fine-tuned models.

Figure 6.11: Average accuracy on sentimental or neutral texts per detection method for all models.



(a) Binary detection.

(b) Discrete detection with includes true cut-off.

(c) Continious detection with above 50% cut-off.

(d) Cot detection.

(e) Discrete detection with above mostly true cut-off.

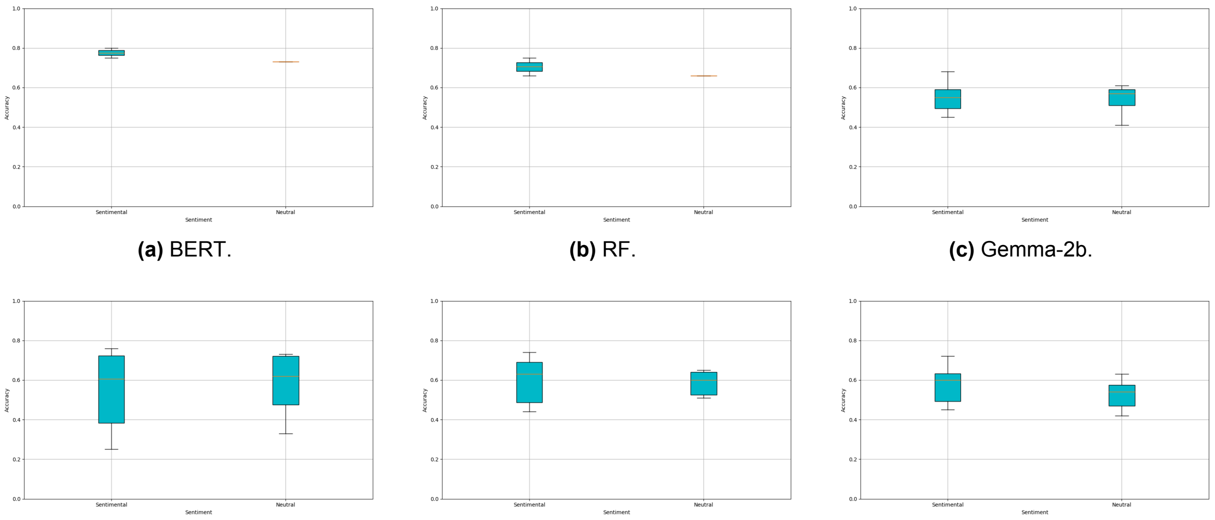
(f) Continious detection with above 75% cut-off.

(g) Binary detection with fine-tuned models.

Figure 6.12: Average misclassifications on sentimental or neutral texts per detection method for all models.

Figure 6.12 illustrates the proportions of false negatives and positives across the different detection methods on sentimental and neutral texts. The results reveal that the binary and CoT detection methods exhibit a slightly higher proportion of false negatives for neutral articles. In contrast, both discrete detection methods show relatively balanced proportions of false negatives and false positives across neutral and sentimental texts. For the continuous detection methods, we notice relatively more false negatives for sentimental articles and more false positives for neutral articles. Finally, the fine-tuned models display a more balanced behaviour, with equal proportions of misclassifications for both neutral and sentimental texts.

A few key trends emerge when examining the average performance of the different models in classifying sentimental versus neutral text (depicted in Figure 6.13). Both the BERT and RF baselines show higher accuracy for sentimental text than neutral text. For the LLMs, although we observe some differences in the placement of the medians, there is generally more overlap between the boxplots, indicating less differences in accuracy between sentimental and neutral text. However, a notable observation is the difference in spread across all models when comparing neutral and sentimental text. Sentimental text typically exhibits a wider spread in accuracy, indicating greater variability in model performance. In contrast, neutral text tends to have a smaller spread, suggesting more consistent classification accuracy across the models. There are no visible differences in the proportions of the models' average false negatives and positives.



(a) BERT.

(b) RF.

(c) Gemma-2b.

(d) Mistral-7b.

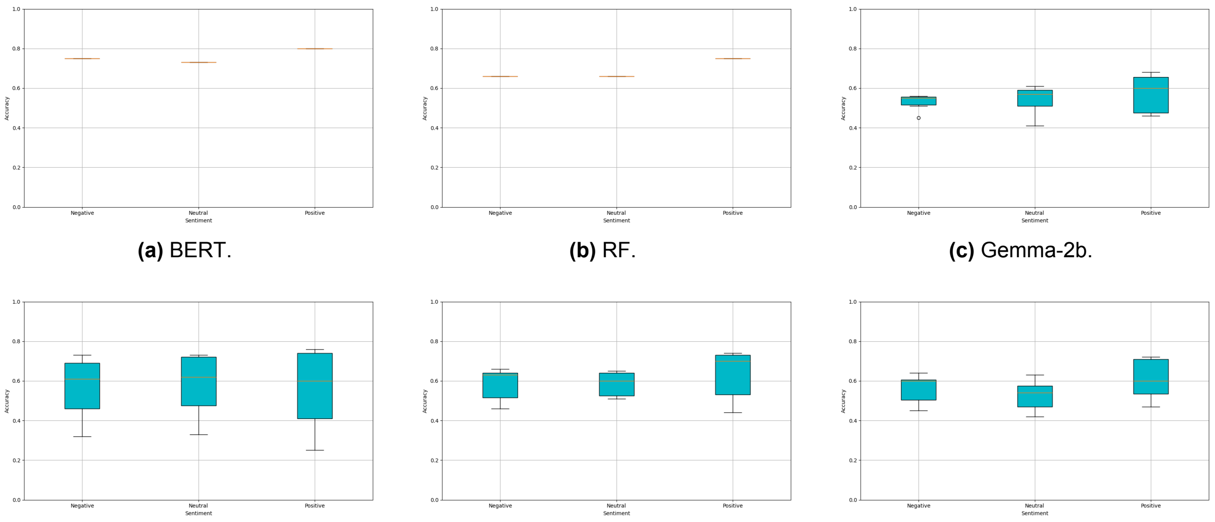
(e) Llama-3.1-8b.

(f) Gemma-2-9b.

Figure 6.13: Average accuracy on either sentimental or neutral articles for all models.

Negative Vs Positive Vs Neutral To further investigate the effects of sentiment, we subdivided the sentimental articles into positive and negative categories. Here, the results reveal more nuanced patterns regarding text sentiment.

When we investigate the impact of the different sentiments on the models' accuracy (shown in Figure 6.15), we observe different patterns for each model. The BERT baseline performs relatively well in classifying positive sentiment, achieving higher accuracy compared to negative sentiment. However, it performs the worst when classifying neutral texts. Similarly, for the Random Forest (RF) classifier, the pattern mirrors that of BERT, with higher accuracy for positive sentiment. However, the model's performance on negative sentiment is slightly lower than BERT's, with worse accuracy on negative texts.



(a) BERT.

(b) RF.

(c) Gemma-2b.

(d) Mistral-7b.

(e) Llama-3.1-8b.

(f) Gemma-2-9b.

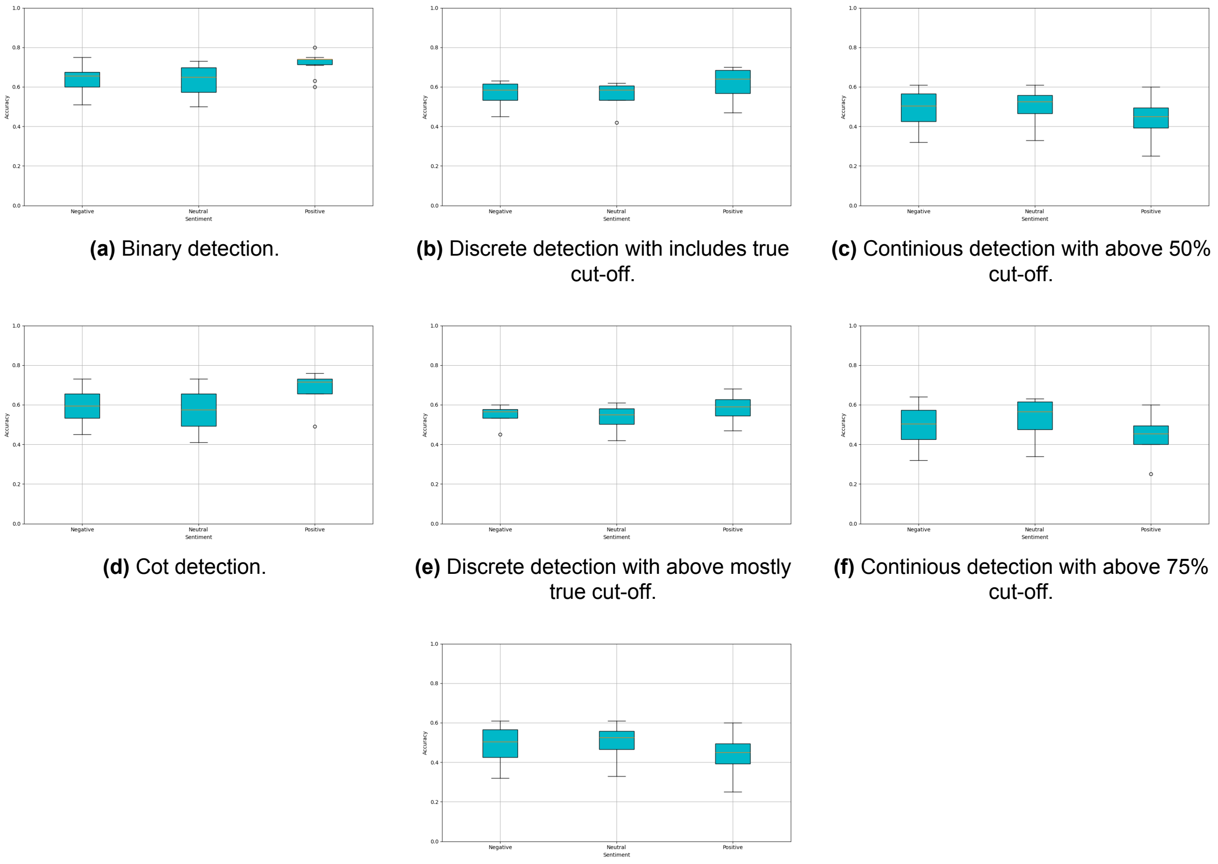
Figure 6.14: Average accuracy on either negative, neutral, or positive articles for all models.

The Gemma model shows a different pattern, with the smallest spread in the boxplot for negative sentiment, indicating more consistent performance when classifying negative texts. The model shows a slightly larger spread for neutral texts and the largest spread for positive texts, suggesting greater variability in performance, especially with positive sentiment articles. For the larger LLMs LLama-3.1, Mistral, and Gemma-2-9b, the medians for accuracy across all sentiments are close to one another, indicating relatively balanced performance across sentiment types. However, both LLaMA and Gemma exhibit smaller spreads in their accuracy distributions than Mistral, indicating more consistent predictions across detection strategies.

For the BERT baseline we see that it is relatively good in classifying the positive sentiment correct in comparison to negative with it being the worst in classifying neutral texts. For the RF classifier we see a similar story but the negative accuracy is a little bit lower. For the Gemma-2b model, we see the smallest spread of the boxplot for the negative text with a slightly bigger spread for the neutral text and the biggest spread for the positive texts. For the bigger LLMs we see the medians being close to eachother but the LLama and Gemma models have smaller spreads.

As with the misclassification for the sentimental neutral split above. No notable results were discovered when investigating the models.

Figure 6.15 depicts the average accuracies across the three sentiment categories for each detection strategy. The binary, CoT, and discrete detection methods all demonstrate higher average accuracy when classifying articles with positive sentiment. In contrast, the continuous detection methods exhibit lower classification accuracy for articles with positive sentiment. This indicates that these models may have more difficulty with positive language. These results suggest that the handling of positive sentiment varies particularly between different detection methods used.



(a) Binary detection.

(b) Discrete detection with includes true cut-off.

(c) Continious detection with above 50% cut-off.

(d) Cot detection.

(e) Discrete detection with above mostly true cut-off.

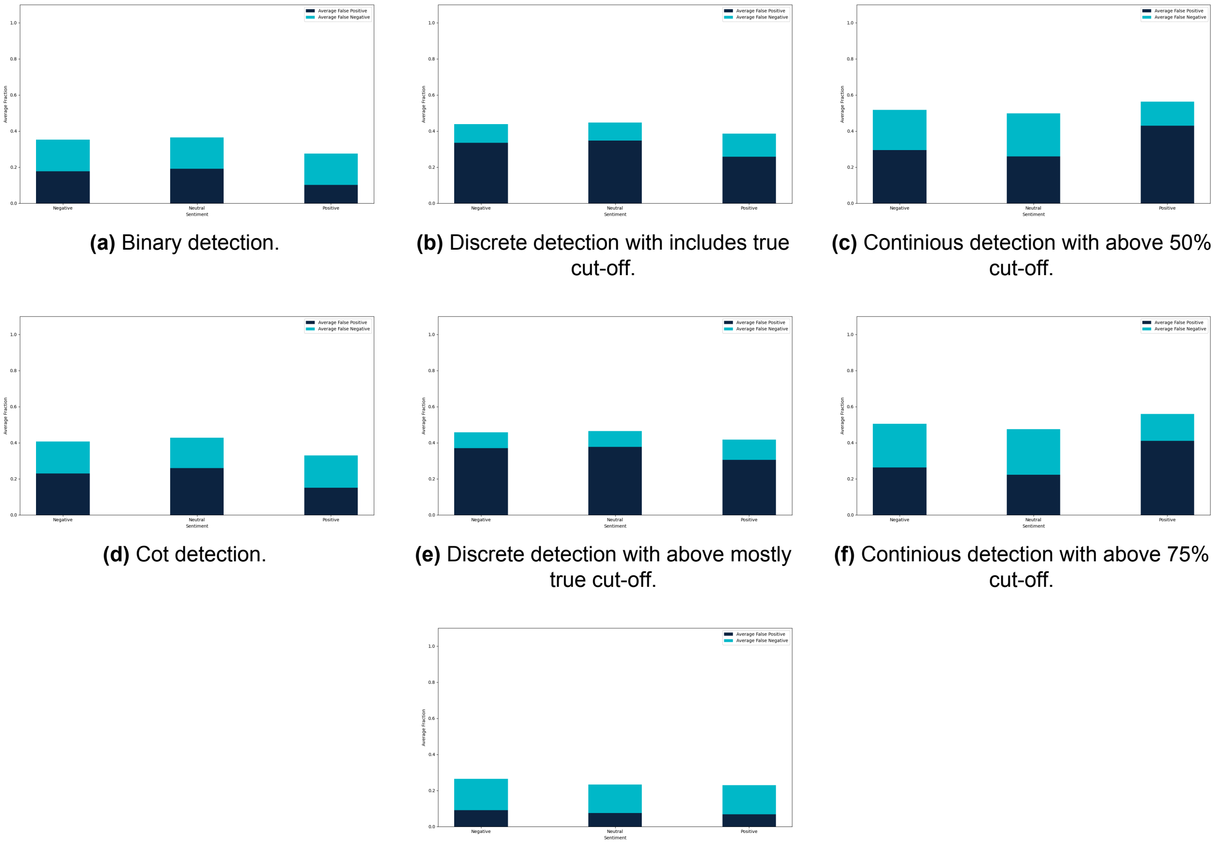
(f) Continious detection with above 75% cut-off.

(g) Binary detection with fine-tuned models.

Figure 6.15: Average accuracy on positive, neutral or negative texts per detection method for all models.

Figure 6.16 illustrates the proportions of misclassifications—specifically false positives and false negatives—across the three sentiment categories for the various detection methods. The results show that for the binary, CoT, and discrete detection strategies, the proportion of false positives is lower for positive sentiments, which helps explain the higher accuracy observed for these models when classifying positive sentiment articles. This reduction in false positives suggests that these models can more effectively distinguish between positive sentiment and other categories, likely due to the more explicit or distinguishable features associated with positive sentiment.

In contrast, for the continuous detection methods, there is an increase in false positives when classifying positive sentiment articles. This increase explains the lower accuracy for positive sentiment in the continuous models, as the models appear to misclassify neutral or negative sentiment texts as positive more frequently. For the fine-tuned models, there is not much variation in misclassifications across the different sentiment categories, once again indicating that these models maintain a balanced performance across all sentiment types. This consistency suggests that the fine-tuned models are better at generalizing across various sentiment patterns, reducing the impact of sentiment on their classification accuracy.



(a) Binary detection.

(b) Discrete detection with includes true cut-off.

(c) Continious detection with above 50% cut-off.

(d) Cot detection.

(e) Discrete detection with above mostly true cut-off.

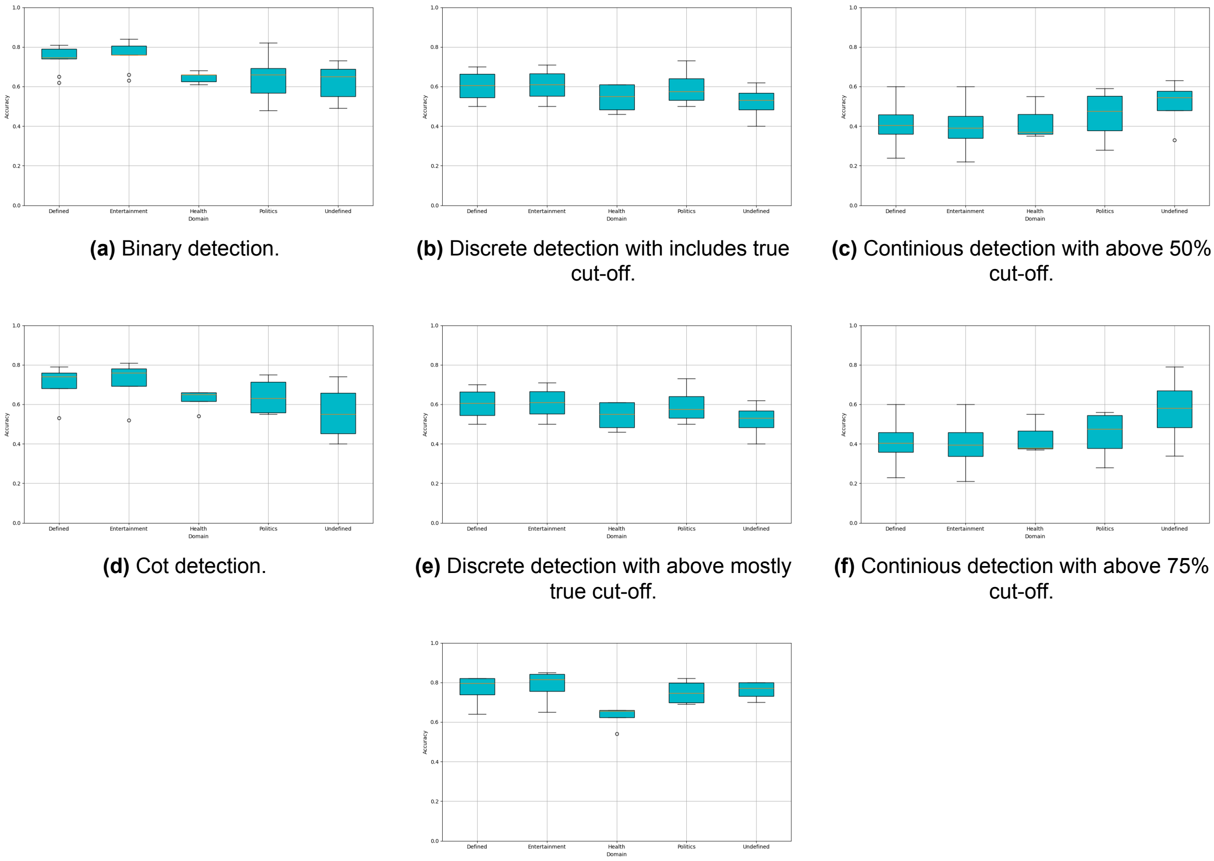
(f) Continious detection with above 75% cut-off.

(g) Binary detection with fine-tuned models.

Figure 6.16: Average misclassifications on negative, neutral or positive texts per detection method for all models.

6.4. Domain

To investigate the LLMs' behaviour through the domain lens, we looked at the different contexts of the news articles. The different datasets we combined have all been labelled with one of the following topics: *Health* from the FakeHealth dataset, *Politics* from the FakeNewsNet PolitiFact split, *Entertainment* from the PolitiFact GossipCop split and finally *Undefined* from the MOCHEG dataset. Because the MOCHEG dataset was intended for a multi-modal Fake News Detection approach and the base dataset also included pictures and videos used to fact-check the claims, we also combined the non *Undefined* contexts into a *Defined* context to see if there was a difference between how the LLMs behave on articles that are fact-checked just on the content in comparison to multi-modal fact-checked articles.



(a) Binary detection.

(b) Discrete detection with includes true cut-off.

(c) Continious detection with above 50% cut-off.

(d) Cot detection.

(e) Discrete detection with above mostly true cut-off.

(f) Continious detection with above 75% cut-off.

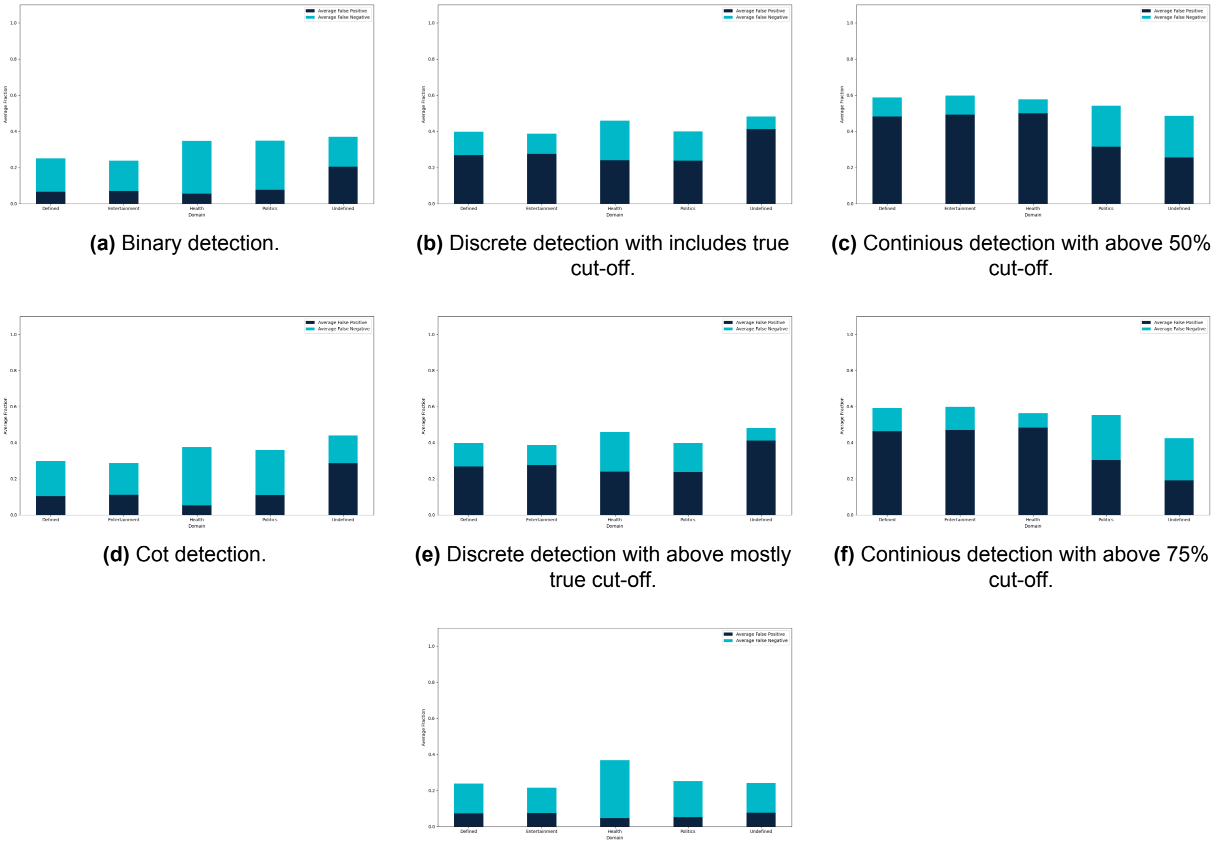
(g) Binary detection with fine-tuned models.

Figure 6.17: Average accuracy of models for different domains per detection method for all models.

We observe the following characteristics when examining the average accuracy of the models per detection method across different domains, as depicted in Figure 6.17.

The binary detection method performs best in the entertainment domain, achieving the highest accuracy. This is followed closely by the politics and health domains, which have similar accuracy levels. Additionally, the binary detection method displays very small spreads in accuracy for both the entertainment and health domains, indicating a high level of consistency in these areas across models. The classifications for the undefined domain show average accuracy that is comparable to the domain of politics, although the domain of politics has a slightly bigger spread, indicating a wider range of variability in accuracy.

The CoT detection method follows a similar pattern to that of the binary strategy. It has the highest accuracy in the entertainment domain, followed by the politics and health domains. However, the CoT model shows much lower accuracy for articles in the undefined domain. For the discrete detection method, the pattern is again similar to that of the binary model, but the entertainment domain displays slightly lower accuracy compared to the other models. In contrast, the continuous detection method presents a different pattern: the accuracy for defined domains is lower than that of the other detection methods, whereas the accuracy of the undefined domain is higher. Finally, the health domain shows the lowest accuracy for the fine-tuned model, indicating that health-related articles are the most difficult for this model to classify correctly.



(a) Binary detection.

(b) Discrete detection with includes true cut-off.

(c) Continious detection with above 50% cut-off.

(d) Cot detection.

(e) Discrete detection with above mostly true cut-off.

(f) Continious detection with above 75% cut-off.

(g) Binary detection with fine-tuned models.

Figure 6.18: Average accuracy of models for different domains per detection method for all models.

When we examine the average misclassifications across the different domains for the various detection strategies (as depicted in Figure 6.18), we see the following trends:

For the binary detection strategy, the health domain exhibits the smallest fraction of false positives relative to false negatives. Across all defined domains (entertainment, politics, health), the binary classification strategy tends to produce more false negatives than false positives on average. However, for articles in the undefined domain, the ratio between false positives and false negatives is more balanced, with no clear dominance of one over the other. The CoT detection method follows a similar pattern to the binary strategy, with a higher proportion of false negatives in defined domains. However, it shows a slightly higher proportion of false positives for articles in the undefined domain.

We see a general trend of more false positives for the discrete detection methods, especially in the undefined domain, which has the highest fraction of false positives among all domains. In the case of the continuous detection method, all defined domains (entertainment, politics, health) have a higher proportion of false positives than false negatives, with the exception of the politics domain, where the distribution of false positives and false negatives is almost even. Notably, the continuous detection strategy has a smaller fraction of false positives overall compared to other detection strategies. For the fine-tuned models, the undefined domain has a lower fraction of false positives than the binary and CoT detection strategies, whereas the distribution of misclassifications of the articles with the defined domain is much more similar.

6.5. Sustainability

To investigate the sustainability lens we keep track of how much energy is consumed during our work. We mainly looked at energy consumption and duration, which were tracked using the *CodeCarbon*[1] Python library. Duration is measured in seconds and energy consumed in kWh. We first discuss the training and fine-tuning processes and then the evaluation, also denoted as inference, of the different models using the various detection strategies. For consistency between results all our experiments were done on the same hardware setup consisting of the following characteristics:

- **CPU**: Intel(R) Xeon(R) Gold 6342 CPU @ 2.80GHz
- **GPU**: 2x NVIDIA A10
- **Location**: The Netherlands
- **OS**: Linux-5.05.0-119-generic-x86-64-with-glibc2.35

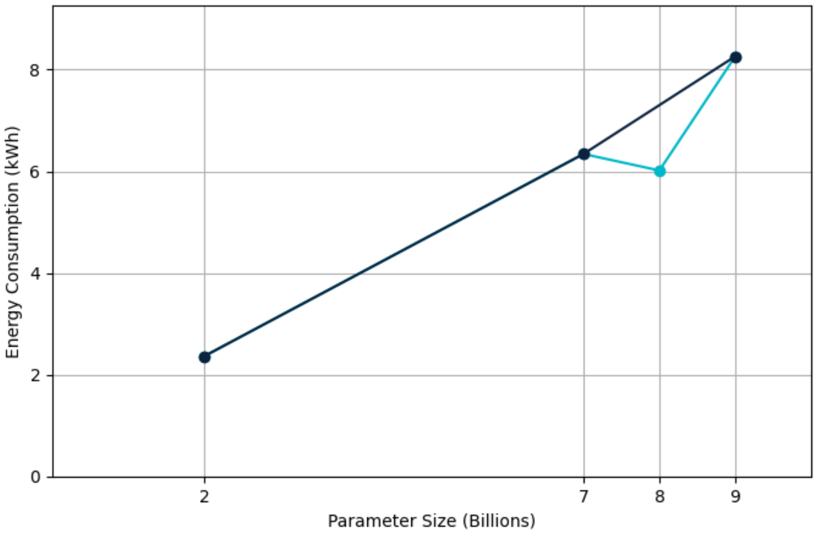


Figure 6.19: Parameter size of LLMs compared to the energy consumption of fine-tuning LLMs.

Training And Fine-tuning Table 6.2 shows the environmental impact characteristics of the training and fine-tuning of the different models. To contextualize the results of the LLMs, we also show the results of training the RF classifier and fine-tuning the BERT model. Compared to the LLMs, the baselines take less training time and consume less energy. The fine-tuned LLMs are trained by using PEFT, specifically LoRa fine-tuning. For the LLMs, a higher parameter size does not necessarily mean a longer duration of fine-tuning. Fine-tuning Mistral, for example, consumes a bit more energy and time than Llama, even though Llama has more parameters. As expected, the smallest model, Gemma-2b, takes the least time to fine-tune and consumes the least energy, whereas the largest model, Gemma-2-9b, consumes the most energy and time. One thing to note is that the results do not show a linear relation between parameter size and energy consumption for the larger models: Gemma-2-9b has 12.5% more parameters than Llama and 29% more parameters than Mistral, but the increase of energy consumption to these models is 37% and 30% respectively. If we then plot the parameter size of the models against the energy consumed for fine-tuning them (Figure 6.19), we see that for Gemma-2b, Gemma-2-9b and Mistral, there does seem to be a linear relation depicted with the dark blue line. When we take LLama-3.1-8b into account (cyan line), we see that this is less strong; this could be explained by LLama being an outlier in terms of architecture.

[1] https://pypi.org/project/codecarbon/

Model	#Parameters	Duration (s)	Energy Consumed (kWh)
Baselines			
BERT	110m	3243.648	0.306
RF	N/A	14.999	0.001
LLMS			
Gemma-2b-it	2b	21104.06	2.37
Mistral-7b-it	7b	56961.48	6.34
Llama-3.1-8b-it	8b	54992.46	6.02
Gemma-2-9b-it	9b	73879.34	8.26

Table 6.2: Environmental impact characteristics of Fine-tuning various models.

Inference Table 6.3 depicts the environmental impact characteristics of inference on the evaluated models. The duration is again measured in seconds, and the energy consumed is in kWh. Because we use the test dataset for some detection strategies and we want to compare them equally, we show the time and energy consumed per 10,000 news articles. As with fine-tuning the models, the baselines take less time and energy for inference, with the RF classifier consuming a negligent amount.

The lowest amount of energy and time consumed for inference for the LLMs is mainly for the binary, discrete and fine-tuned detection strategies. Mistral shows that the percentage-based inference is comparable to that of the binary, discrete and fine-tuned models, whereas, for the other models, the percentage-based detection strategy consumes more energy and time. The CoT detection strategy consumes the most energy for all models compared to the other detection strategies, almost a magnitude higher in three models. It is also evident that a higher impact on the environment does not necessarily mean a higher accuracy, as for all the models, the fine-tuned strategy outperforms CoT, while CoT consumes much more energy and time.

Model	Accuracy	Size Dataset	Duration (s)	Energy Consumed (kWh)
Baselines				
BERT	0.77	6416	150.80	0.01
RF	0.71	6416	0.70	0.00
LLMS				
Gemma-2b-it				
Binary	0.55	32077	1828.82	0.20
Binary	0.57	6416	1546.79	0.17
Fine-tuned	**0.65**	6416	2342.11	0.26
CoT	0.44	32077	**16831.19**	**1.83**
Discrete Mostly True	0.47	32077	1613.50	0.18
Discrete True	0.63	32077	1613.50	0.18
Percentage 50	0.51	32077	2162.28	0.25
Percentage 75	0.52	32077	2162.28	0.25
Mistral-0.2-7b-it				
Binary	0.72	32077	4281.33	0.52
Binary	0.72	6416	4282.37	0.51
Fine-tuned	**0.81**	6416	4801.13	0.56
CoT	0.63	32077	**38712.11**	**4.94**
Discrete Mostly True	0.54	32077	4325.58	0.52
Discrete True	0.61	32077	4325.58	0.52
Percentage 50	0.28	32077	4324.71	0.52
Percentage 75	0.28	32077	4324.71	0.52
LLama-3.1-8b				
Binary	0.7	32077	3944.28	0.49
Binary	0.7	6416	4239.83	0.53
Fine-tuned	**0.7**	6416	4884.07	0.57
CoT	0.67	6416	**37154.66**	**4.63**
Discrete Mostly True	0.55	32077	3964.28	0.49
Discrete True	0.66	32077	3964.28	0.49
Percentage 50	0.46	32077	7231.28	0.90
Percentage 75	0.47	32077	7231.28	0.90
Gemma-2-9b-it				
Binary	0.48	32077	4822.21	0.54
Binary	0.54	6416	4896.66	0.56
Fine-tuned	**0.8**	6416	6143.15	0.67
CoT	0.6	6416	**65222.71**	**3.80**
Discrete Mostly True	0.39	32118	4390.84	0.50
Discrete True	0.45	32118	4390.84	0.50
Percentage 50	0.6	32118	6978.81	0.79
Percentage 75	0.61	32118	6978.81	0.79

Table 6.3: Environmental impact characteristics of inference on various models. *Note: duration and energy consumed are per 10,000 articles*

7 Discussion

In this chapter, we describe our findings and contextualize them with earlier research. We discuss our results through various lenses and discuss the limitations of our work.

7.1. Discussion through the different lenses

This section describes the findings from the different lenses we investigated, which we use to answer research question 3. We aim to give analysis, possible explanations and implications of the results described in chapter 6. It is, however, hard to give particular answers about why the LLMs behave in the ways they do because of our lack of information about architecture, training data and the inner workings of the LLMs. Nevertheless, we have gained some interesting insights.

Performance In our evaluation, we analyzed the performance of the models across the following metrics: valid label predictions, accuracy, and the occurrence of false negatives/positives. Each of these metrics plays a distinct role in assessing how effective and practical a model is for automated fake news detection.

The fraction of valid label predictions is important for determining how easily a model can be integrated into a large-scale fake news detection system. As discussed earlier, this metric refers to whether the models' predictions align with the expected output format, such as "fake" or "real". While a binary classification model might produce labels like "This news is mostly true", these kinds of predictions could be problematic if the system is designed to parse only binary classifications. Therefore, even if these classifications are correct individually, they may not be usable in a system that expects specific labels. Hence, models that generate a high fraction of valid labels are more easily adjustable for automated systems.

Simultaneously, accuracy, along with the types of misclassification, is crucial for evaluating the model's actual performance in detecting fake news. Even if a model produces a valid label, if it is incorrect, the system becomes ineffective for real-world applications. A model that produces only valid labels but has low accuracy and many false positives and negatives would still result in a poor system for automated fake news detection. False positives can undermine the credibility of a system, whereas false negatives can result in the continued spread of fake news.

The base models achieved a perfect fraction of valid label predictions, with 100% of their predictions being valid. This is understandable, given that they were specifically designed to give only one of two binary classifications. In contrast, the LLMs do not always produce valid labels. For all models, the binary classification methods (binary standard, CoT and fine-tuned) show instances where the fraction of valid label predictions was imperfect and, in some cases, even fell below 80%. This can be attributed to the difference in how different models respond to different prompts.

In our preliminary testing, before finalizing the prompts used in our evaluation, we already noticed variations in how well the models responded to certain prompts. This variation suggests that the effectiveness of prompting is highly model-dependent. Some LLMs may struggle to consistently output valid labels, especially when trained to give nuanced answers.

These findings highlight the importance of carefully designing prompts, both for the specific task at hand and the model that is leveraged. This customization is critical for ensuring the LLMs can be reliably used in automated systems.

Compared to the baselines, the base versions of the LLMs do not achieve the same level of accuracy. This is explainable by the fact that both baselines are specifically trained for the task of fake news detection, allowing them to optimize for this particular classification task. In contrast, the base versions of the LLMs are more general models and have not been fine-tuned for this task, which explains their lower accuracy. However, when fine-tuned, some of the models outperform the baselines, although this improvement is not guaranteed. In our evaluation, only the Mistral and Gemma-2-9b models managed to achieve a higher accuracy than BERT's 77%, reaching 81% accuracy. This indicated that while fine-tuning can greatly improve performance, not all models benefit equally from it.

A notable pattern that emerged during our evaluation is that, on average, the continuous detection methods exhibit the lowest accuracy. This can be explained by the nature of the LLMs. They are trained on vast amounts of natural language to find patterns to predict the next words. Finding patterns about which number comes after a given task is a much harder problem than just predicting the next word. When discrete detection methods are employed, we observe a slight improvement in accuracy, though it still remains lower compared to other detection strategies. This improvement over the continuous detection strategy can be explained by the fact that discrete methods provide natural language options instead of numerical outputs. As a result, the discrete detection method aligns more with the LLMs' innate patterns.

The false positives were generally much higher with the discrete detection method than with the other detection strategies. This is quite an interesting phenomenon because, in the prompt, there are as many options to classify something as fake as there are to classify it as real. Something that could explain why more of the predictions of the models using the discrete detection method predict the fake half of the discrete options could be that the LLMs somehow give more importance to the earlier options described in the discrete prompt. Another interesting phenomenon of the discrete detection method is that it generally has the highest amount of valid answers compared to the other strategies. This could be because the base models are trained with data that includes possible responses in arrays.

The next best-performing detection strategies are the binary classification methods. The basic binary and CoT binary classification methods achieve similar accuracy levels on average. However, we observe that CoT prompting can occasionally result in higher accuracy, though in some cases, it also leads to lower accuracy. This variability can come from differences in the reasoning capabilities of the models. Some models are better equipped to handle the reasoning processes required by CoT prompting, while others may struggle with more complex reasoning chains, leading to inconsistent performance when averaged over the different models. Another factor contributing to these variations is that different models perform better with specific types of CoT prompting. While we explore various prompting strategies in our study, the models' effectiveness depends heavily on how they were prompted. Given the diversity in model behaviour, investigating optimal CoT strategies for each model could form the basis of a separate study.

The fine-tuned models performed best for the bulk of the LLMs. This can be attributed to the fact that only they are specifically trained for fake news detection. Interestingly, for some models, such as LLama and Mistral, the fine-tuning does not increase but decreases the number of valid labels predicted. For all models, valid accuracy is higher when fine-tuned.

Looking at the different models, we also see some interesting results. For one, a bigger model does not necessarily mean it is more accurate or valid. For example the biggest model: Gemma-2-9b has the lowest validity of all models using binary classification and even the valid accuracy was lower than that of the 7b parameters Mistral and 8b parameters LLama model. As expected, the oldest and smallest model, Gemma-2b, performed worst on validity and accuracy. This can mostly be attributed to its high number of false positive predictions, which, in turn, can be attributed to its relatively small parameter size. There is also no clear relation between model release date and performance: Mistral is the oldest model in our evaluation and still outperforms most other models on most detection strategies. If we try to find the best model for fake news detection while keeping in mind their size and age, it seems that Mistral wins, especially when fine-tuned, which is impressive given that it is the only one not coming from a giant tech behemoth such as Google and Meta.

Textual Characteristics We investigated the impact that textual characteristics of news content can have on leveraging LLMs by looking at two characteristics. These characteristics are text length and readability. Text length of the news articles can not only impact the classifications of these models by the amount of sources and context that is given in an article, but it can also be that LLMs have a more challenging time “remembering” what was discussed earlier in the article when settling on its classification. Readability is an important metric because complicated language can be interpreted as more thorough and valid reasoning. We measure readability by using the Dale-Chall score, with a higher score indicating that the text is harder to interpret.

When investigating the impact of text length on the different detection strategies we do not see a notable difference in accuracy along different lengths. However, when evaluating the two types of misclassification, we see a notable pattern in the binary, CoT, and discrete detection methods. We see a shift around the 50% text length mark for both types of misclassification. With the number of false positives first increasing and then decreasing and the number of false negatives first decreasing and then increasing. This is especially interesting when contextualising this with the data analysis done in Section 5.3. Here, we see that, in general, the fraction of real news increases with text length, which would expect us to see fewer instead of more false negatives for text length, assuming that the models are not impacted by text length. This discrepancy could indicate that the LLMs are prone to label longer texts as real, even when their content is not. Bad actors could exploit this trend to fill their news with additional content to make it longer and get around the ability of LLMs to detect it.

For the continuous detection method, it can be observed that as text length grows, the number of false positives increases, and the number of false negatives decreases. This aligns more with the dataset’s innate characteristics. Finally, the fine-tuned models depict no notable differences in terms of classification regarding text length. This indicates that fine-tuning helps mitigate the innate trends of misclassification based on the text length that the models exhibit.

As with the results for the different detection strategies, the evaluation of the various models does not show any notable differences in terms of accuracy regarding different text lengths, while they follow the same patterns regarding misclassifications.

When evaluating the impact of readability on the different detection strategies, we did not observe patterns as notable as those seen with text length. There are small differences in accuracy across the various detection methods but no significant trends. For the binary, CoT, and discrete detection strategies, there appears to be a slight increase in accuracy around a readability score between 6 and 7. This suggests that the models perform slightly better when dealing with texts that are neither simple nor too complex. Interestingly, some sources[1] suggest that the reading level of the average United States citizen is around 7th and 8th grade, corresponding with a Dale-Chall score between 6 and 7. Given this context, it can be theorized that because LLMs are trained to understand the average person’s input, they are better at understanding language at the average person’s literacy level. Making them also better at detecting fake news on this level. The results of misclassifications regarding readability correspond with those of the accuracy.

Psychology Prior research in communication theory and fake news suggests that the psychological features of text play an important part. This can be better understood by investigating the predictions of the models through this lens. We first evaluate the models’ classifications by comparing sentimental texts (positive and negative combined) with neutral texts, assessing how the models handle emotionally charged content versus more objective texts. Following this, we compare model performance across the more specific sentiment categories: positive, neutral, and negative.

Evaluating the different detection strategies when comparing neutral vs. sentimental texts, the results imply that neutral text is harder to classify for the binary, CoT, and discrete detection methods. This can be explained by the higher accuracy of these detection strategies when classifying positive news. This higher accuracy stems from a lower amount of false positives for positive text compared to the other sentiments. This could imply that the base LLMs have developed a more substantial alignment with positive texts.

Contrastingly, for the continuous detection strategy, results show a relatively higher accuracy for neutral text stemming from a higher misclassification rate on positive text. It is not clear why the different detection methods differ so much in their ability to accurately classify text with positive sentiment. It does mean that the ability for these models to correctly classify text of different sentiments is substantially impacted by the detection strategy employed. When fine-tuned, the predictions of the LLM differ less between the sentiments.

[1] https://web.archive.org/web/20170309032834/http://literacyprojectfoundation.org/community/statistics/

When evaluating the different models through different sentiments we do not see much difference in average accuracy or misclassification rates, but there can be a difference in spread observed. This is likely due to the difference in accuracy for the various detection strategies described above.

Domain The impact of the domain of news to correctly classify the veracity of news is important because, depending on the domain, the impact spread of fake news can vary greatly. This lens is evaluated by investigating the predictions of the different models on five domains. Defined (including health, entertainment and politics) and undefined, of which the domain was unknown.

Investigating the different detection methods, the results show for the binary, CoT and discrete strategies that undefined news is more likely to be misclassified. Specifically, a higher fraction of false positives is found. This is likely due to the nature of the MOCHEG dataset of which all data is labelled as undefined. The MOCHEG part of the data was the only part that was initially fact-checked with accompanying videos and pictures. Because these media aspects of the articles are not used for the evaluation of the models, this can lead to incomplete information for the LLMs to take into account.

The CoT detection strategy performs worse on the *Politics* domain. This is particularly interesting because the CoT prompting technique is guided to use the same reasoning structure that PolitiFact uses for manual fact-checking in the context of political news. When we look at the False positives for the CoT detection strategy for predictions in the domain of politics, we see that compared to the other binary classification strategies, there are more False negative predictions, meaning that the CoT prompting technique more often labels political news as true when it is actually fake.

For the defined domain, the results show that the health domain is the hardest to classify accurately for binary, discrete and CoT detection strategies. Investigating the types of misclassification for the health domain, there is an overrepresentation of false negatives. This could stem from the fact that to effectively detect fake news in the health domain, a broad amount of specific knowledge is required. This inability to detect fake health-related news is especially worrying because of the negative implications it can have on society. Sadly, even with fine-tuning, the LLMs still perform worse on classifying health-related news. The continuous detection strategy is better for classifying the veracity of news in the undefined and health domain.

Sustainability The results of the LLMs, as investigated through the sustainability lens, imply the following findings. First, larger parameter sized models are not necessarily worth in terms of accuracy gain compared to the amount of energy they consume and therefore the amount of emissions they produce. For example, the Gemma-2b model consumes about a third of the amount of energy that the Gemma-2-9b model does but when looking at their raw accuracy (not valid accuracy) we see that Gemma-2b still outperforms its larger and newer variant.

We also see that even though leveraging a CoT detection strategy does generally improve accuracy it also consumes substantially more energy. Another practical implication we found is that as long as data is split uniformly over different domains, sentiments, and text lengths, the models behave the same for smaller as for bigger datasets. That is to say, the models are consistent enough in their predictions that when doing this kind of investigation, researchers will not need a complete dataset to produce reliable results. Research and industry can reduce their emissions by testing and analyzing their strategies for a small dataset.

7.2. General Discussion

In this work, we evaluate LLMs from various lenses and use several detection methods to better understand how to leverage them for fake news detection. Now that we have discussed the results and various lenses, we answer the research questions, highlight the main takeaways, and give some practical recommendations.

We find many similarities and differences with our diverse set of LLMs evaluated. One such similarity is that, more often than not, fine-tuning improves their capability to classify and detect fake news. These fine-tuned models also have some of the highest scores, even compared to the baselines. This shows that there is real potential for LLMs to detect fake news effectively. We also find that, in general, there is not much difference between different cut-off values for the discrete and continuous detection methods.

This points towards the idea that although they can give predictions that seem nuanced, they might not be. If there is a bigger difference between different cut-off values, for example, that would indicate that the models are more capable of giving nuanced predictions. We have not seen any evidence of this happening.

Another interesting trend we see is that CoT prompting does not necessarily make the model more accurate, but it does make them consume much more energy and time. We have seen a difference in performance between the smallest LLM, Gemma-2b, and the larger models in terms of accuracy, with the larger LLMs scoring higher. However, we do not see a big difference between the size of the vendor of the LLMs and their performance. Google and Meta are two tech giants, whereas Mistral is relatively small but still the best-performing LLM in our work. We also do not see a big difference in accuracy based on the release date between the LLMs. Even though Mistral is also the oldest model, it was also the one with the highest accuracy.

The main takeaway we find when trying to answer research question 1 is that performance differs greatly based on which LLMs are employed and which detection strategies are used. Regarding the size of the LLMs, the number of parameters matters up until a certain point. The best-performant LLM through all lenses was Mistral when fine-tuned, which was substantially bigger than the Gemma-2b (the smallest model) but smaller than the other two LLMs evaluated. CoT prompting can help, but it needs to be tailored correctly for the specific model when used for fake news detection. Additionally, fine-tuning LLMs makes them more performant while costing less energy than CoT prompting.

The RF classifier and BERT baseline show better performance than most models with most detection methods. This is especially interesting because they are much easier to deploy in terms of hardware requirements, quicker to train and faster to run. Compared to the base models, the RF classifier only has less accuracy than the Mistral base model. Additionally, the BERT baseline is only beaten in performance by the fine-tuned versions of Mistral and Gemma-2-9b. Fine-tuning does, however, not always improve the fraction of valid predictions of the output. This makes it harder to leverage LLMs consistently for fake news detection, a problem from which the baselines do not suffer. It is important to note that for the baselines, we use two models that are already specifically trained on the data, whereas the base LLMs are trained to follow natural language instructions; they are possibly not trained on fake news detection specifically. In the end, LLMs can outperform earlier PLM and feature-based models, but they need to be leveraged in the right way.

Answering research question 2, we find that compared to base LLMs, more straightforward automated fake news detection strategies seem to be better. Both feature-based strategies and pre-trained language models perform better than base LLMs and cost substantially less energy to train and use for inference. However, if the goal is to focus solely on the performance of fake news detection and disregard the impact on the environment, then fine-tuning LLMs is the better choice.

The results and insights gained make it clear that LLMs fare differently from the different perspectives discussed. LLMs are slightly impacted by text length, misclassifying longer articles more often as true, which can be exploited by bad actors. In terms of readability, there is no big difference between readability and how LLMs classify articles, but there is a small tendency to better classify articles with readability corresponding to that of a U.S. citizen in grade 7 or 8. This could imply that LLMs can also be suitable for fact-checking social media posts, where there are possibly more people writing with an average reading level than in the news. LLMs are generally better at classifying sentimental news which is a good sign given that fake news often plays on the emotions of their readers. A worrying trend we see is that in the context of health news, the LLMs perform worst. After a worldwide pandemic and with more and more people adopting LLMs as an alternative to web search, this can lead to even bigger problems for society relating to fake health news.

Finally, compared to earlier models, the LLMs are way less sustainable and the impact on the environment is much higher, even without taking into account the initial training to develop the base models. When not the highest amount of accuracy is needed for fake news detection, the earlier methods, such as using feature-based classifiers of PLMs, are the better choice. There are, however, ways to reduce the environmental impact when using LLMs for fake news detection, especially when doing research and testing. When evaluating or training a LLM for fake news detection, one can first reduce the dataset to a smaller version, given that the characteristics stay the same. We did this by applying a stratified split to our final dataset of around 32.000 news articles, giving us our test set, and saw no notable differences regarding average accuracy over all detection strategies between evaluation on the different dataset sizes.

Then this dataset can be fine-tuned in a sustainable manner such as PEFT, in our case we used LoRa, and then finally evaluate and test the models.

7.3. Limitations

In this section, we describe the limitations of our work. From data-related to model-related and limitations inherent to fake news detection itself.

Data There has been much work done in collecting and verifying labelled datasets for fake news detection, and it is remarkable that by combining earlier research, we have a total dataset of more than 30.000 labelled news articles to evaluate the various models. There is, however, still a sparsity in fake news datasets in the domain of *Politics*, which is arguably much more important to mitigate than that of, for example, *Entertainment* news. This sparsity could have led to less generalizable results in our evaluation. The datasets are also relatively old from an information technology perspective. It could well be that some of the articles then labelled fake are now real and vice-versa, which would make our results less reliable because of mislabeled ground-truth values.

Another possible limitation of our work is the fact that we do not know what exact training data the models evaluated are trained on. This opens up the risk for possible data contamination in the sense that the models already “know” that some news is fake a priori. We tend to think that if there were a lot of data contamination, the LLMs would perform much better than they do now (in the 90% range instead of a maximum of 80% now).

Also, as described in the section above, the MOCHEG dataset was initially composed for multi-modal fake news detection, whereas we only use a content-based detection strategy. Earlier tests of us when experimenting with the pipeline did indicate that the base and fine-tuned models performed better on just the FakeHealth and FakeNewsNet datasets. Finally, the deeper underlying question still needs to be asked: What makes News true or false and what is true and false? Although it seems that this question might be impossible to answer, it does matter a great deal about how much we can truly trust the base datasets.

Openness The problem of Openness of LLMs is still an important limitation in our work. We can sadly only theorize about why the models behave like they do. Without precise descriptions about their architecture and training data we cannot know for certain why LLMs generate the predictions that they do. As the field of LLMs evolves and grows, this issue will become more and more problematic as long as there are no significant efforts made in explainable AI.

Prompting Because of our limited hardware resources, we were forced to make some concessions in how we leveraged the LLMs for fake news detection. For one, we had to load the models in as 16-bit floats instead of the standard 32-bit floats. This means that we have been working only with half-precision potentially significantly decreasing performance over all metrics. This was done to decrease computation time and memory usage, which are both heavily reduced when using lower-precision floats. Additionally, to reduce memory usage, we also had to cut off the longer types of articles to a smaller text length when prompting to not get an out-of-memory error. This could again impact the prediction results of the models. These limitations do, however, not invalidate our research because even with half-precision, the models performed comparable to and, in some cases, even outperformed our baselines.

Fine-tuning For fine-tuning, we had to make similar concessions as with prompting in the sense that we needed to cut off some of the lengthy articles, and we loaded our models for training in 8-bit precision (even lower than those used for prompting) because of hardware limitations. Another limitation of our fine-tuning was that we did little work in optimizing the hyperparameters and training settings. We did some preliminary testing and made sure that we had a correct data split for training, testing and validation. However, we did not extensively optimize the fine-tuning. This was mainly because we were more interested in the general gain of fine-tuning over different models and is intended to be used for future research. These limitations mainly imply that there is probably still a lot of optimization to be gained and put our results on the more conservative side of the potential of LLMs for fake News detection.

Part V

Closure

8 Future work

In this manuscript we set an informed precedent for how to leverage LLMs for fake news detection. From this work future research can expand on the insights gained here and push the domain of automated fake news detection forward.

As new LLMs are being developed and introduced to the LLM ecosystem these new models can be evaluated and fine-tuned using our framework and methodology. Additionally, larger models should be investigated. We evaluated the smaller range of models between 2b and 9b parameters, although bigger versions already exist at the time of writing this work. For example, Google's Gemma has a version of 7B parameters, Gemma-2 has a version of 27b parameters, Mistral has a new model with 22b parameters, and Meta has a Llama-3.1 model with 405B parameters.

As our work suggests that there does seem to be an improvement of models with larger parameter size. With these larger models, there could also be a deeper evaluation be performed on loading the models in the different float types. We were hindered by hard ware limitations and had to make concessions in this regard, but with enough compute power these would be interesting experiments.

Additionally, more investigation could be done specifically for CoT and Binary classification prompting which reached the highest accuracy of valid predictions in our evaluation. We did apply general known guidelines to our prompt engineering, but we have not done an in depth optimization. This would have required us to find for each model the optimal prompt and would making comparisons much harder and generalizable.

Our fine-tuning work could be expanded upon in future work. We used a very generic LoRa fine-tuning method mostly because of hardware limitations, but full-fledged fine-tuning should also be explored. As we did not seek to introduce new models, our emphasis on fine-tuning should be more seen as a conceptual grounding of the positive effect that fine-tuning can have on LLMs for fake news detection. Future work could do an in-depth analysis on how to best fine-tune LLMs for fake news detection.

For our evaluation we used a diverse and large dataset although relatively old in terms of news content. High quality datasets are hard to collect and evaluate, but future work could introduce new datasets that could be used for future evaluations using our frame work and methodology.

As earlier works also focused on other parts of the SMCR model, such as the sender, receiver and context, future work could combine strategies based on these factors with our work to improve automated fake news detection. One example, could be to combine social network information on how fake news information is disseminated on social media platforms with an LLM based content prediction technique as ours akin to the work done by Wu et al. [37].

To further analyze the effect that domains of articles have on LLM predictions, more work could be done to investigate entities most prevalent in articles for both types of misclassification predictions. Specifically, in certain domains such as *Politics* and *Health*, work could be done to find potential biases against certain political ideologies or certain medical practices. For example, an investigation could be performed on whether models are maybe misclassifying news about vaccinations more than about antibiotics.

Other psychological features could be explored to increase understanding of how the psychological features of text affect the veracity classifications of LLMs. One such feature could be the emotions exhibited in news articles, such as joy, anger, and sadness.

Finally, new techniques such as Retrieval Augmented Generation (RAG) could be used to do in-depth domain specific fact-checking. This technique could fact-check statements with up-to-date information from reputable sources. This could potentially greatly improve the fake news detection capabilities of LLMs by not only using their general knowledge and reasoning capabilities, but also ground them in sources and truth. An added benefit of such an approach would also be the ability for the users of the fake news detection systems to validate and further research predictions of the LLMs. Which is currently very hard to do.

9 Conclusion

This manuscript explored the effectiveness, opportunities, and challenges of employing LLMs for the automated detection of fake news, a critical task in the ongoing effort to mitigate the societal impact of disinformation. Through a systematic evaluation, we addressed three key questions regarding the performance of LLMs in comparison to traditional methods, differences between LLM architectures, and the results observed through multiple lenses. With this, we contributed a holistic evaluation of LLMs for fake news detection and set a precedent for reproducible and reliable future work in this realm.

Our findings demonstrate that LLMs offer promising advancements in fake news detection, outperforming two traditional approaches (feature-based classification and PLM-based fake news detection) when fine-tuned. However, the performance of LLMs varies greatly depending on the model used, which generation strategy is employed and the level of guidance given. Although some of the fine-tuned models perform well, others do not. Certain models give only valid predictions, whereas others do not, making them harder to use in combination with automatic systems parsing their classification. The baselines provide a good contextualization for the evaluation of the LLMs by showing that earlier techniques are still viable and have a much smaller impact on the environment. Applying multiple lenses to evaluate these models helps reveal how LLMs' strengths and weaknesses manifest across different perspectives, such as textual characteristics, psychology, domains and sustainability. We find trends in the LLM predictions that can be exploited by bad actors such as a propensity to classify longer texts more often as real. LLMs also have a harder time accurately classifying neutral texts, which could potentially be exploited. Further, we also found a clear weakness of the LLMs, one of which is that our results show that LLMs more often than not classify health-related news as true, which can further the spread of conspiracy theories about vaccinations and medicine and reduce trust in medical professionals. It is clear that in terms of sustainability the LLMs are a worse option compared to earlier methods that use exceptionally less energy when leveraged. How the models are leveraged can also have a massive impact on their carbon footprint. CoT prompting was shown to be the worst in terms of time needed and energy spent, and a more cost-effective option is to use PEFT followed by the use of simple binary detection with the newly fine-tuned model.

Despite these promising results and takeaways, our work still has limitations. Issues such as a lack of computing power, the availability of diverse datasets, and the openness of LLMs present ongoing challenges. Although we produce the first work using these multiple lenses to get a holistic understanding, there is still much work to be done and many perspectives to be explored.

Our key contributions include a better understanding of how to employ LLMs for fake news detection, their strengths and weaknesses, and some practical recommendations for deploying them in the real world. Additionally, we provide a systematic approach that describes not only how to evaluate LLMs but also past and future fake news detection systems using a multiperspective methodology. Finally, we also share our code, where other models and datasets can be easily incorporated for future analysis and evaluation.

In conclusion, while LLMs hold great potential for advancing automated fake news detection, thoughtful consideration of their limitations and careful refinement of their application are essential for their effective deployment in the fight against fake news. As millions are being invested in LLM applications, fake news detection is not an area where we can take shortcuts and apply them without thought and consideration. By building upon these insights, researchers and industry practitioners can contribute to more accurate and reliable methods for safeguarding the integrity of information in the digital age and, with that, ensuring a safe and trustworthy shared reality for us all.

References

[1] Salman Bin Naeem et al. "An exploration of how fake news is taking over social media and putting public health at risk". In: *Health Information & Libraries Journal* 38.2 (2021), pp. 143–149. DOI: https://doi.org/10.1111/hir.12320. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/hir.12320. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/hir.12320.

[2] Jenifer Whitten-Woodring et al. "Poison If You Don't Know How to Use It: Facebook, Democracy, and Human Rights in Myanmar". In: *The International Journal of Press/Politics* 25.3 (2020), pp. 407–425. DOI: 10.1177/1940161220919666. eprint: https://doi.org/10.1177/1940161220919666. URL: https://doi.org/10.1177/1940161220919666.

[3] H. Ardiff. *A Report of the Detailed Findings of the Independent, Office of the United Nations High Commissioner for Human Rights*. Retrieved from https://policycommons.net/artifacts/3446961/a/4247098/ on April 1, 2024. Office of the United Nations High Commissioner for Human Rights, 2018.

[4] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[5] David MJ Lazer et al. "The science of fake news". In: *Science* 359.6380 (2018), pp. 1094–1096.

[6] Onur Varol et al. "Online Human-Bot Interactions: Detection, Estimation, and Characterization". In: *Proceedings of the International AAAI Conference on Web and Social Media* 11.1 (May 2017), pp. 280–289. DOI: 10.1609/icwsm.v11i1.14871. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14871.

[7] Sijing Chen et al. "Spread of misinformation on social media: What contributes to it and how to combat it". In: *Computers in Human Behavior* 141 (2023), p. 107643. DOI: https://doi.org/10.1016/j.chb.2022.107643. URL: https://www.sciencedirect.com/science/article/pii/S0747563222004630.

[8] D.K. Berlo. *The Process of Communication: An Introduction to Theory and Practice*. The Process of Communication: An Introduction to Theory and Practice v. 10. Holt, Rinehart and Winston, 1960. URL: https://books.google.nl/books?id=0k9IAAAAMAAJ.

[9] Nir Kshetri et al. "The Economics of "Fake News"". In: *IT Professional* 19.6 (2017), pp. 8–12. DOI: 10.1109/MITP.2017.4241459.

[10] Felipe Ortega et al. "On the Inequality of Contributions to Wikipedia". In: *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*. 2008, pp. 304–304. DOI: 10.1109/HICSS.2008.333.

[11] Shervin Minaee et al. *Large Language Models: A Survey*. 2024. arXiv: 2402.06196 [cs.CL]. URL: https://arxiv.org/abs/2402.06196.

[12] Jason Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 24824–24837. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[13] Shunyu Yao et al. "Tree of thoughts: Deliberate problem solving with large language models". In: *Advances in Neural Information Processing Systems* 36 (2024).

[14] Potsawee Manakul et al. "Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models". In: *arXiv preprint arXiv:2303.08896* (2023).

[15] Noah Shinn et al. "Reflexion: Language agents with verbal reinforcement learning.(2023)". In: *arXiv preprint cs.AI/2303.11366* (2023).

[16] Sarah J Zhang et al. “Exploring the mit mathematics and eecs curriculum using large language models”. In: *arXiv preprint arXiv:2306.08997* (2023).

[17] Tongshuang Wu et al. “Promptchainer: Chaining large language model prompts through visual programming”. In: *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 2022, pp. 1–10.

[18] Yongchao Zhou et al. “Large language models are human-level prompt engineers”. In: *arXiv preprint arXiv:2211.01910* (2022).

[19] Zeyu Han et al. *Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey*. 2024. arXiv: 2403.14608 [cs.LG]. URL: https://arxiv.org/abs/2403.14608.

[20] Edward J Hu et al. “Lora: Low-rank adaptation of large language models”. In: *arXiv preprint arXiv:2106.09685* (2021).

[21] Gemma Team et al. *Gemma: Open Models Based on Gemini Research and Technology*. 2024. arXiv: 2403.08295 [cs.CL].

[22] Gemma Team et al. “Gemma 2: Improving open language models at a practical size”. In: *arXiv preprint arXiv:2408.00118* (2024).

[23] Stella Biderman et al. “Pythia: A suite for analyzing large language models across training and scaling”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 2397–2430.

[24] Zhengzhong Liu et al. “Llm360: Towards fully transparent open-source llms”. In: *arXiv preprint arXiv:2312.06550* (2023).

[25] Albert Q. Jiang et al. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL].

[26] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL].

[27] Abhimanyu Dubey et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[28] Josh Achiam et al. “Gpt-4 technical report”. In: *arXiv preprint arXiv:2303.08774* (2023).

[29] Mykhailo Granik et al. “Fake news detection using naive Bayes classifier”. In: *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. 2017, pp. 900–903. DOI: 10.1109/UKRCON.2017.8100379.

[30] Benjamin Horne et al. “This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news”. In: *Proceedings of the international AAAI conference on web and social media*. Vol. 11. 1. 2017, pp. 759–766.

[31] Anu Shrestha et al. “Textual characteristics of news title and body to detect fake news: a reproducibility study”. In: *European Conference on Information Retrieval*. Springer. 2021, pp. 120–133.

[32] Yaqian Dun et al. “Kan: Knowledge-aware attention network for fake news detection”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 1. 2021, pp. 81–89.

[33] Jian Cui et al. “Meta-Path-based Fake News Detection Leveraging Multi-level Social Context Information”. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. CIKM ’22. Atlanta, GA, USA: Association for Computing Machinery, 2022, pp. 325–334. DOI: 10.1145/3511808.3557394. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3511808.3557394.

[34] Van-Hoang Nguyen et al. “FANG: Leveraging Social Context for Fake News Detection Using Graph Representation”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. CIKM ’20. Virtual Event, Ireland: Association for Computing Machinery, 2020, pp. 1165–1174. DOI: 10.1145/3340531.3412046. URL: https://doi.org/10.1145/3340531.3412046.

[35] Enyan Dai et al. “Ginger cannot cure cancer: Battling fake health news with a comprehensive data repository”. In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 14. 2020, pp. 853–862.

[36] Erxue Min et al. “Divide-and-Conquer: Post-User Interaction Network for Fake News Detection on Social Media”. In: *Proceedings of the ACM Web Conference 2022*. WWW ’22. <conf-loc>, <city>Virtual Event, Lyon</city>, <country>France</country>, </conf-loc>: Association for Computing Machinery, 2022, pp. 1148–1158. DOI: `10.1145/3485447.3512163`. URL: `https://doi-org.tudelft.idm.oclc.org/10.1145/3485447.3512163`.

[37] Jiaying Wu et al. “Prompt-and-align: prompt-based social alignment for few-shot fake news detection”. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2023, pp. 2726–2736.

[38] Chenxi Whitehouse et al. “Evaluation of fake news detection with knowledge-enhanced language models”. In: *Proceedings of the international AAAI conference on web and social media*. Vol. 16. 2022, pp. 1425–1429.

[39] Mateusz Szczepański et al. “New explainability method for BERT-based model in fake news detection”. In: *Scientific reports* 11.1 (2021), p. 23705.

[40] Menglin Liu et al. “PoliPrompt: A High-Performance Cost-Effective LLM-Based Text Classification Framework for Political Science”. In: *arXiv preprint arXiv:2409.01466* (2024).

[41] Xiaofei Sun et al. *Text Classification via Large Language Models*. 2023. arXiv: `2305.08377 [cs.CL]`. URL: `https://arxiv.org/abs/2305.08377`.

[42] Yazhou Zhang et al. *Pushing The Limit of LLM Capacity for Text Classification*. 2024. arXiv: `2402.07470 [cs.CL]`. URL: `https://arxiv.org/abs/2402.07470`.

[43] Ye Liu et al. *Detect, Investigate, Judge and Determine: A Novel LLM-based Framework for Few-shot Fake News Detection*. 2024. arXiv: `2407.08952 [cs.CL]`. URL: `https://arxiv.org/abs/2407.08952`.

[44] Kai Shu et al. “Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media”. In: *Big data* 8.3 (2020), pp. 171–188.

[45] Barry Menglong Yao et al. “End-to-end multimodal fact-checking and explanation generation: A challenging dataset and models”. In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2023, pp. 2733–2743.

[46] Anu Shrestha et al. “Textual characteristics of news title and body to detect fake news: A reproducibility study”. In: *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*. Springer. 2021, pp. 120–133.

[47] Benjamin Horne et al. “This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news”. In: *Proceedings of the international AAAI conference on web and social media*. Vol. 11. 1. 2017, pp. 759–766.

[48] Gyeong-Geon Lee et al. “Applying large language models and chain-of-thought for automatic scoring”. In: *Computers and Education: Artificial Intelligence* 6 (2024), p. 100213.

[49] Paul Cobley et al. *Theories and models of communication*. Vol. 1. Walter de Gruyter, 2013.

[50] Jeremy Howard et al. “Universal Language Model Fine-tuning for Text Classification”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych et al. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 328–339. DOI: `10.18653/v1/P18-1031`. URL: `https://aclanthology.org/P18-1031`.

[51] C J Hutto et al. “VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text”. In: *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*. Ann Arbor, MI, 2014.