Eberhard Karls Universität Tübingen
Seminar für Sprachwissenschaft

# Master Thesis

**Approaches to the automatic detection of machine-generated text**

Winkler Aron

October 2024

**Reviewers**

Prof. Dr. Çağrı Çöltekin
Seminar für Sprachwissenschaft
Universität Tübingen

Prof. Dr. Gerhard Jäger
Seminar für Sprachwissenschaft
Universität Tübingen

**Winkler Aron:**
*Approaches to the automatic detection of machine-generated text*
Master Thesis
Eberhard Karls Universität Tübingen
Thesis period: July - November 2024

## Abstract

Recent developments in Natural Language Processing (NLP) have resulted in the development and popularization of highly effective Large Language Models (LLMs), capable of generating convincing and seemingly creative linguistic material. LLMs have garnered much attention, both from researchers and the general public, and continue to be increasingly applied to a variety of fields. However, the breakneck speed at which these systems are adopted leaves unattended some of the security concerns regarding their use. Since the language produced by the models is of such high quality, it is not always feasible to distinguish authentically human contributions from machine-generated text, which can enable a multitude of nefarious applications of LLMs. This work explores the history and inner workings of LLMs, how they can be misused, and possible antidotes to the problem of machine-generated text detection, with a careful eye toward a good balance of computational cost and performance of detection strategies.

## Acknowledgements

## Contents

# 1 Introduction

Recent advances in the field of language generation have unlocked the door to the widespread use of language models, across almost all fields of human life – academics, marketing, arts, programming, and so forth. For anyone engaging with digital media, talk of generative artificial intelligence (AI) has become a daily occurrence, and many companies are integrating language technology in some form or another. As it stands, chatbots like ChatGPT and image generators like DALL-E are used by millions to make art, tell exciting stories, improve customer relationships, create tailor-made learning environments for students, and much more. In this panorama of exciting new technologies, however, there is no lack of worrying signs, and potential for the misuse of these innovative tools.

Language generation has been around since last century, so its massive popularization in 2020 wasn't necessarily as groundbreaking for the discipline as it was for the general public. What represented the biggest leap was the quality of the output that the language models were capable of. The generation of material largely indistinguishable from human-written text was available for everyone to use through popular interfaces like ChatGPT. Given the scale at which AI-generated content has been flooding into all manners of digital phenomena, it has become increasingly crucial to develop up-to-date technologies to detect when a piece of media (for the purposes of this work, only text is considered) is authentically human, or machine-generated.

Language generation has a huge number of positive applications, and it stands to improve people's lives in many ways. Still, the potentially nefarious uses are equally as many. In fact, even without bad intent, it is easy for model operators to generate and use harmful language in some form.

One example of language models empowering malicious actors is phishing and scamming. These attacks involve attempting to make their victims reveal sensitive information, or perform certain actions (for example, send money to the attacker or reveal their login information to some service). Phishing thus far has been (fortunately) plagued by scaling issues – to increase the number of attacks, one had to hire new human attackers, which is expensive and time-consuming. With language generation, it could become possible to bypass this limitation by assigning some (or even most) of the work to automated systems. As models grow more sophisticated, they might become even better suited to gain people's trust for an attacker to exploit, or trick them out of sensitive information outright.

The information landscape, already shaky in the digital age of social media, could also suffer from the malicious employment of language generation. Disinformation campaigns in the political, social, and commercial spheres have been commonplace throughout human history, but, similar to phishing, have been notoriously expensive to scale. The ability to generate highly convincing content to further disinforming claims is a great boon to those who seek to spread fabrications and widen divisions. Worryingly, AI is already being shown to suit the intent to *dis*inform particularly well, to the point that it may even work better as a tool for deception rather than truth.

One need not even intend to harm when employing NLG systems to inadvertently cause damage. Plagiarism, as an example, is a very real danger with language models, since at any point they may generate one-to-one reproductions of their training data. On the same note, if the model's training data contained biases and misinformation – either naturally or as a result of adversarial manipulation – the generated text could further these without obvious signs or explicit intent. Chapter 3 contains further elaboration on these issues, which were only superficially mentioned in this introduction.

In the past, automatic generation of text material relied on relatively simple technologies, and was thus comparatively easier to detect. The ability to accurately flag generated text could be relied upon with pre-LLM systems, and the quality was often insufficient for the more sophisticated tasks. Being alerted to the use of AI lessens its danger in most cases – an inflammatory article is far less effective, a scam email far less convincing, and fake homework much less troublesome when the AI authorship can be reliably exposed. Unfortunately, starting with GPT-3, not only has the quality of model output grown considerably, but even human evaluators have a tough time separating LM generations from authentic human productions. Needless to say, automatic detection systems have also suffered a decrease in effectiveness and have had to upgrade their arsenal to keep up.

This thesis aims to delve into the field of machine-generated text detection, exploring the current state of the art, as well as limitations and future possibilities. Outside of analyzing the various approaches to the problem, this work is meant as a contribution to the development of systems that do not forgo all size concerns in favor of performance. Modern detection systems rely on models whose running cost is comparable to the massive language models doing the generation – but the performance gains of this strategy do not necessarily justify the tradeoffs.

When deploying computationally heavy solutions, it is often assumed that they will not run on the end user's machine – it is after all unreasonable to expect the average user to have hundreds of Gigabytes of RAM on hand. Instead, the client software is only a relay to the centrally hosted service, with which it communicates over the network. This may be acceptable in some cases, but it is not desirable in others – for example, users may and should be reluctant to dispatch their private communications to some remote service in the name of detecting generation.

Fortunately, as with many things, system design is a game of tradeoffs with performance on the one end and complexity on the other. Highly complex systems, relying on huge models to make predictions, will likely be the best detectors of machine-generated text. This fact has been true in the field as well as in other areas of computational linguistics: state of the art models often take several GPUs to train and run, with high associated cost. Still, it is often the case that 80% of performance can be salvaged at only 20% of the cost. This work is written with the somewhat substantiated belief that a version of this proves true in designing solutions for machine-generated text detection as well.

There are many strategies that can be employed to detect machine generation at a reasonable computational costs. Some examples include pretrained embeddings, which are a valid substitute of contextual embeddings provided by LLMs, and linguistically motivated features, an classical representation of text information that can be computed cheaply. Task-specific models and even language models are still useful tools. Not all models are huge, and there is fruitful research of transformer-based models like DistilBERT, which maintain performance close to the original with far fewer parameters. These methods are discussed in higher detail in Chapter 4.

Target selection is of course important, but customer-facing system designers should aim for standalone solutions that would be able to run on mid-to-high-end machines, when this is feasible. In this work, a series of options to achieve this are presented, with some of them directly experimented with in the context of Task 8 at SemEval 2023. The proposed models to detect LM generation employ, among other strategies, GPT-2-based perplexity metrics, linguistically motivated features, pretrained embeddings, contextual embeddings with small models, and task-specific character-level models. A full discussion of the methods is presented in Chapter 5.

Note to professor: I plan to write 2-3 more paragraphs here to conclude the introduction more elegantly, but I haven't been able to come up with a version I like yet.

## 2 Background

In 1951, Gnome Press published Isaac Asimov's *Foundation* (Asimov, 1951), the first title of a trilogy that would go on to become one of the cornerstones of modern science fiction. In the novel, set in the distant future, scientist Hari Seldon predicts the fall of the Galactic Empire, an event that would pave the way to an era of barbarism in the story's fantastical universe.

To preserve humanity's knowledge and technical skills, Hari Seldon establishes the Foundation on an uninhabited planet on the periphery of the Empire, a sort of outpost dedicated to being the home to the archival effort. The novel follows the political and technological adventures of the Foundation and its leaders, with one of the first plot points being the first conflict between the Foundation and a major local power in the periphery, Anacreon, which declared its independence as the Empire's influence in the periphery weakened. Seeking protection from the Empire against Anacreon's expansionary stance, the Foundation hosts a diplomatic emissary from the Empire, a Lord Dorwin, finally obtaining a convoluted treaty between the Empire and Anacreon over their respective spheres of influence.

> "Before you now you see a copy of the treaty between the Empire and Anacreon – a treaty, incidentally, which is signed on the Emperor's behalf by the same Lord Dorwin who was here last week – and with it a symbolic analysis."
> The treaty ran through five pages of fine print and the analysis was scrawled out in just under half a page.
> "As you see, gentlemen, something like ninety percent of the treaty boiled right out of the analysis as being meaningless, and what we end up with can be described in the following interesting manner:
> "Obligations of Anacreon to the Empire: None!"
> "Powers of the Empire over Anacreon: None!"
>
> *Isaac Asimov, Foundation,*
> *Part II: The Encyclopedists*

At this point the Foundation's scientists, through a technique they call *"symbolic analysis"*, condense several pages of treaty into a few lines, revealing the hidden meaning behind the layers of legal dissimulation. By doing so, they expose the inability of the dying empire to exert its influence over its own periphery, and they realize that moving forward, they can only rely on themselves, marking perhaps the true starting point of the story in Asimov's *Foundation*.

Despite first reading this passage when I was a teenager, perhaps over a decade ago, these fictional twists stayed with me through the years. They were, after all, my first indirect exposure to the field of computational linguistics and natural language processing (NLP). I remember being mesmerized by the potential of machine computation applied to natural language, in what I would later learn to better define as a mixture of information retrieval and automatic text summarization.

While Asimov's pen definitely hit the mark in predicting some of the most intriguing and successful applications of NLP in the years ahead, what granted computational linguistics perhaps its brightest moment in the limelight was one of its other, albeit related, subfields: language modelling and generation.

## 2.1 Language modelling

Teaching a machine to understand and produce natural language is intuitively a difficult task. Even if one could reliably collect all ingredients that make up human language, creating a system that emulates it even just well-enough is a very tall order, since there would likely be millions if not billions of cases to consider. Linguists have documented hundreds of languages, each with their own grammar, peculiarities, exceptions, all of which have yet to be described under one common ruleset. Manually building a program from the ground up for even just one language is beyond what current technology is capable of.

The very first chatbot, ELIZA (Weizenbaum, 1966), simulated conversation through pattern matching and substitution, essentially repeating and paraphrasing their interlocutor's statements. While it successfully bypasses the necessity of programming a machine with *intelligence*, such an approach does not result in a system that can be described as creative in any sense. In other words, ELIZA will never write a poem, or surprise their conversation partner with a witty turn of phrase. It would never be able to tell whether May has 30 or 31 days because it has no notion of what *May* and *days* are. Teaching language is, after all, not only an issue of grammar, but one of world knowledge as well.

If *teaching* language to machines as one would to humans is not possible, and rule-based approaches such as ELIZA inevitably reach a bottleneck, then it becomes necessary to adopt a new strategy, rooted in statistics. This new approach consists in the realization that the sentence "he's wearing a circumference jacket" is much less likely to be uttered than "he's wearing a yellow jacket". Extrapolating the pattern, the set of words that can fill the gap in "he's wearing a _____ jacket" is varied, but "yellow" will have a much higher *probability* of showing up than "circumference". Language models are the tools that are employed to estimate these probabilities.

Due to recent innovations in NLP, the phrase "language model" evokes big and expensive systems, trained on huge amounts of data and costing enormous amounts of money to develop. While this is certainly understandable, the label in itself has no presupposition of size or cost. In essence, language models, and NLG (Natural Language Generation) systems in general, break down the massively complex problem of "teaching language to machines", into the more manageable task of "statistically learning what words are likely to follow others". In other words, language models produce next-word (or, more generally, next-token) probabilities based on an input sequence (Gao and Lin, 2004). After this is achieved, the resulting model can be prompted over and over, one word at a time, in order to generate long pieces of text, by a process called autoregressive generation (Lin et al., 2021).

For example, for the completion "fifteen minutes of _____", one would expect a (good) language model to offer words such as "fame" or "overtime". One idea to achieve this is to collect some linguistic data and observe what words follow "fifteen minutes of" and extrapolate a probability distribution from the observed frequencies. So-called *n-gram* language models (Chen and Goodman, 1999) are built in this fashion, with the $n$ in *n-gram* specifying the amount of left context taken into consideration.

The simplest of these models, the bigram (2-gram) language model, records co-occurring word pairs in the sample dataset. Consequently, for this model, only the last word of a sequence determines the prediction over the following word. This results in a model that can reliably generate short collocations, such as "Marie *Curie*", but cannot generate coherent sentences, and would likely even fail to offer "fame" as a completion to "fifteen minutes of _____", since the only available context for the prediction is the word "of". To correctly predict "fame", one would need at least a 4-gram language model, which would finally allow for such a "long" context requirement. However, while taking more context tokens into

consideration increases the performance of n-gram language models, it does so at a steep (especially memory) cost: for 4-gram LMs with a vocabulary size (i.e., how many words the model knows) of 1000, for example, implementations without optimizations would require the frequency counts for $10^4$ n-grams to be accessible for predictions.

Another issue that presents itself with frequency-based models is the handling of unseen n-grams. For example, the 3-gram "duck goose pony" may never come up in the model training data, in which case some near-0 probability is assigned to the sequence (due to smoothing, see for example Chen and Goodman (1999)). While some n-grams will fail to appear due to being ungrammatical or nonsensical like in "duck goose pony", others may be perfectly well-formed but either rare or just absent from the training data due to chance: if "trees need hydration" was never observed, then the model would fail to recognize this n-gram to be more likely than, for example, "trees need circumference" (assuming the latter was also not observed, which is quite likely in organic text). Even the n-gram "roundly faucet knowledge" would be equally as likely as "trees need hydration" if both were never observed in training. While the latter example can be solved by including lower-order n-grams in the probability calculation ("trees need" may have been observed even if "trees need hydration" wasn't, but "roundly faucet" is unlikely to have been observed; see Katz (1987)), the former case requires more subtle knowledge. In order to correctly assess "trees need hydration" as a quite likely n-gram, the model would need to identify the similarity between the words "water" and "hydration", and infer that "trees need hydration" should have higher probability than, say, "trees need virtual", due to "water" and "hydration" being similar.

Due to such limitations, n-gram language models are not the piece of technology that propelled language modelling to the heights that we associate with it today. The missing piece of the puzzle are neural networks (Anderson, 1995), which when applied to language generation give rise to *neural language models*.

Following the example above, both n-gram and neural language models solve the fundamental problem of estimating the probability that the word "fate" follows "fifteen minutes of". However, in order to do so, n-gram language models draw upon explicit frequency observations when generating its output, an approach that often fails to consider an adequate amount of context, or to take into account the fact that similar words appear in similar contexts. In contrast, neural language models draw upon their internal parameters - the weights and biases (Anderson, 1995) associated to the various layers of the neural network that makes up the model.

## 2.2 Neural language modelling

Neural networks are an extremely powerful for many applications across several disciplines. Providing an effective summary of all neural networks is a difficult task, since they manifest themselves in different variations for different tasks. Still, they are generally understood as interconnected layers of *neurons* that map an input vector of numbers to an output vector. Each neuron in the network is made up of a vector of weights, a bias, and an activation function, which are used to transform an input vector to an output number. For the purposes of this work, it is more important to understand the overall network rather than its individual parts: neural models are a way to apply complex transformations to vectors. The *parameters* of the model, i.e. the combined weights and biases of the individual neurons, can be used to encode knowledge in a way that simpler statistical models struggle to achieve.

Above, the example of "trees need water" and "trees need hydration" was briefly discussed. While n-gram models have no structural way to note the similarity between "water" and "hydration", and thus fail to recognize that the admissible contexts for the two words have

some overlap, neural networks have been employed to solve this problem exactly because of their capacity to progressively store knowledge. Word embeddings (Selva Birunda and Kanniga Devi, 2021) are a way of representing words as numerical vectors, such words with similar semantics will be close to each other in terms of vector distance. The embeddings for "water" and "hydration" would therefore be closer in vector space than "water" and "dog". Several ways have been developed to derive embeddings from text data — for example, the CBOW (Mikolov et al., 2013) algorithm uses neural networks to predict the missing word given the surrounding context through the use of a neural network. For many iterations, the model is presented with a piece of text with a gap, and the objective of guessing the missing item. With each example, the model parameters are updated, or *nudged* in the correct direction (for information on gradient descent, see Zhang (2019)), i.e. towards a state that are more conducive to the correct prediction. Importantly, among the parameters of the neural network are the embeddings for every word, which are used by the model to compute the final prediction across subsequent layers — these are random at the beginning, but progressively more and more refined. At the end of training, most model parameters are discarded, but the embeddings are kept, and hopefully the result will have captured semantic similarities between the entries.

While the CBOW algorithm discards all model parameters aside from the embeddings, it is naturally possible to train the neural model with the objective of keeping all of them, still taking advantage of the architecture's capacity to learn and store information. This is the basis for modern, highly sophisticated language models, with billions or even trillions of parameters.

## 2.3 Large Language Models

Language models have a long and intricate history, having been iterated upon from different perspectives and with different architectural approaches. The introduction of the transformer (Vaswani et al., 2023), a model type that for more efficient training over extremely large text data, propelled language model quality forward considerably, to the point that all language models commonly known today follow this architecture. The ability to train models with relatively little expenses allowed models to grow further and further in perplexity, eventually resulting in what we identify today as Large Language Models (LLMs). In terms of research attention, BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) have perhaps been the most resonant examples earlier on.

Bidirectional Encoder Representations from Transformers (BERT) is slightly different from the language models discussed so far, in that its primary purpose is not generation. The word embeddings described above are a powerful tool for obtaining meaningful representations, but have the significant flaw of not being context-aware. For example, it's clear that the word "honey" should have different embeddings between "bees make honey" and "honey, wake up!".

BERT is a solution to this problem: instead of training a model to then only keep the embedding layer (in other words, computing all embeddings *offline*), BERT is a full-fledged language model that evaluates pieces of text as a whole to return their representations *online*. As such, when evaluating "bees make honey" and "honey, wake up!" with BERT, "honey" will have vastly different embeddings to account for the different context.

BERT was developed, in other words, to compute context-aware word representations, hence the name. It cannot be used for language generation, mostly because it's nature as a *bidirectional* model. For the example language model architectures described above, there was an underlying assumption that only the *left* context is visible to the model. This makes

intuitive sense: when generating language, only what came before influences the probability distribution of the next word. This is not the case for BERT. Since the objective here is to evaluate finished productions, BERT may use all previous and subsequent tokens at each timestep.

Despite being a fairly recent introduction to the scene, being made available in just 2018, BERT has made big waves in nearly all fields where processing natural language is even tangentially relevant. It even resulted in the birth of the somewhat jokingly named discipline of BERTology, which is meant to convey the detailed analysis of how BERT and its different variations exactly arrive at the high-quality embeddings that they are known for.

In the same year as BERT was published, another historical language model also made its debut: OpenAI's GPT-2. GPT-2 (Radford et al., 2019) lines up closer to the popular idea of what constitutes a language model. It is large in size with 1.5 billion parameters, and was primarily conceived for language generation. In the original paper, the authors highlighted the ability of the model to approach several different problems without explicit training, such as question answering, test summarization, machine translation, and so forth. In this sense, GPT-2 is one of the first successful examples of language models displaying generalized problem-solving skills, that require both articulation and minute world knowledge.

GPT-2 has become more of a baseline than a challenger in the years following its introduction, such was its impact in research applications. It has garnered much attention and analysis, similarly to BERT, a process that also highlighted some of its flaws (see, for example, the GPT-2 unicorn story[1]). While talk about GPT-2 could not at all be considered undertone, it was perhaps cut short, in that nowadays it is more rarely employed compared to BERT. This is in large part due to its new iteration, GPT-3 (Brown et al., 2020).

## 2.4 Fears and reactions to LLMs

In 2020, GPT-3 was announced by the company OpenAI, and was made available to the general public through an interface called ChatGPT. With its introduction, the gates were open to the generation of high-quality text through AI. This was perhaps the first example of language model garnering tremendous general attention, not only in academic circles but from the public at large. ChatGPT even experienced service outages due to its servers not being able to handle the astounding traffic they were receiving. The introduction of GPT-3 provided an unprecedented boost to language models as a consumer technology, leading to a sort of arms race both in integrating AI into customer-facing products, as well as in model development itself. In the first wave of AI competition, Facebook's Llama (Touvron et al., 2023) and the open source model Mistral (Jiang et al., 2023) also entered the scene, alongside Google's now discontinued Bard (see for example Fowler (2023)). In a subsequent wave, further developments have reached even higher generation quality, with models such as GPT-4 (OpenAI et al., 2024) and Gemini (Team et al., 2024).

Such developments marked the beginning of language models being commonplace technology. The arts, education, technology, sales, and dozens of other fields have seen major changes to the way they operate, either in alongside or in opposition to artificial intelligence. Education, in particular, was one of the first areas to experience a problematic application of language generation, with a prevailing initial fear that students would opt to have a language model solve their homework for them. Surprisingly, programming also saw the rapid birth of assistive technology, such as Github Copilot (Chen et al., 2021), leading to fears that many

[1] `https://pbs.twimg.com/media/DzYpsJOU0AA1PO9.png:large`

software professionals would be made obsolete by the new technologies.

People employed in artistic fields, particularly writers, also took a deeply cautious stance from the beginning with respect to AI. In a landmark development, the 5-month-long strike of the Writer's Guild of America[2] resulted in an agreement which included safeguards for writers against the use of artificial intelligence[3]. Such a stance turned out to be far from unfounded, with how AI-generated content has flooded several Internet platforms. As recently as March 2024, even the Google search engine has had to address the issue of unoriginal content, which is in large part meant to target results which contain generated text[4]. At the same time, the search engine itself adopted a new feature called AI Overviews, an integrated AI-generated response to the user's query[5].

The use of large language models thus has been observed to spark mixed reactions, both emotionally and legislatively. However, what was considered so far in this examination is the *lawful* and (perhaps arguably) moral use of language technology. This does not mean that such technologies cannot be employed with harmful intent — quite the contrary. the discussion about malicious use of language generation has been left on the sidelines in the early years of LM adoption, but worries are progressively making their way to the forefront of the debate, and countermeasures are becoming increasingly pursued research objectives.

## 3 Threats posed by NLG systems

NLG systems, and language models in particular, have emerged as an extraordinarily useful tool in a number of creative and technical fields, but it stands to reason that they would lend themselves to nefarious applications just as well as ethical ones. Following Crothers et al. (2023), several threat models (Shostack, 2014) have been proposed to tackle the potential dangers of language generation, some of which have already had real-life realizations as well.

Threat modelling provides researchers and professionals with the tools to predict potential dangers related to resources or technologies, even (and especially) in absence of historical expertise related to the relevant threats. For example, when designing a messaging application, a possible exploitation attempt could involve a malicious actor eavesdropping on other users' conversations. A threat model designed around this scenario would try to define the characteristics of the attacker, the likely strategies used, and how to counter them, preemptively or reactively. The methodical evaluation of potential threats and the implementation of relevant protection mechanisms is of particular concern in the field of information technology, since this area is so synonymous with scalability. Indeed, when something goes wrong in systems that operate on a huge scale, as do many if not most digital services people interact with daily, the impact is often proportionally disastrous.

NLG systems integrate themselves into this picture better than one might initially assume. Importantly, language models scale far more easily than a human force. Humans need to be taught, few at a time and for a period of time, to perform a task. Automated systems, however, have no such limitation: once trained, one must only increase the compute to

[2] An article summarising the strike can be viewed at https://www.vox.com/culture/2023/9/24/23888673/wga-strike-end-sag-aftra-contract

[3] See for example https://www.nbclosangeles.com/news/local/hollywood-writers-safeguards-against-ai-wga-agreement/3233064/ for an account of the agreement

[4] Google published a blog post explaining the changes at https://blog.google/products/search/google-search-update-march-2024/

[5] Google published a blog post explaining its new AI integration at https://blog.google/products/search/generative-ai-google-search-may-2024/

increase coverage, an operation that takes far less time and resources than training people.

As such, language generation makes the known universe of digital threats that much more severe. On the one hand, it boosts the scale at which already existing attacks were operating, such as scams and phishing attempts - both of which had to be performed by human actors in some way thus far. On the other hand, it creates entirely new threats which were previously unfeasible, such as generating convincing documents for homework-evasion.

## 3.1 Scam and phishing attempts

Attacks and exploitation attempts targeted at individuals are particularly well-suited for NLG usage, since they usually involve using natural language to convince the victim of performing some action, such as revealing sensitive information or granting access to a restricted resource.

Phishing is a cyberattack method where attackers impersonate legitimate entities to trick individuals into revealing private information, such as passwords, credit card numbers, or personal details. This is typically done through deceptive emails, messages, or websites that appear trustworthy but are actually fraudulent. Victims are often lured into clicking malicious links or downloading harmful attachments, leading to the theft of their data or the compromise of their devices. Phishing is a widespread and dangerous tactic used to gain unauthorized access to systems and conduct identity theft, financial fraud, or other malicious activities.

An example of a phishing attack could involve a fake email that appears to come from a well-known bank. The email might use the bank's logo, official-sounding language, and a convincing sender address to create a sense of urgency, such as warning the recipient that their account has been compromised. The message instructs the recipient to click on a link to verify their account information immediately. When the user clicks the link, they are directed to a counterfeit website that looks almost identical to the bank's real site. Once on this fake site, the victim is prompted to enter their login credentials, which are then captured by the attackers. With this information, the cybercriminals can access the victim's real bank account, potentially leading to financial loss and identity theft.

NLG has been proven to be effective across several varieties of phishing attacks. Depending on the scope of the attack, one can distinguish phishing attacks targeting indiscriminate groups (massive phishing), specific communities (community phishing) or individuals in particular (spear phishing).

Regarding the former, the exploitation attempt targets a specific group or community, such as members of a social club, employees of a particular company, or even residents of a neighborhood. The attacker leverages the shared characteristics, interests, or affiliations of the community to create a more convincing and believable scam. For instance, a cybercriminal might send an email that appears to come from the community's leader, such as a club president or company CEO, announcing an upcoming event or an urgent matter requiring action. The message may ask recipients to click on a link to RSVP, pay dues, or access important information. This link, however, leads to a fraudulent website designed to capture login credentials, financial details, or other sensitive information. By exploiting the trust and familiarity within the community, these phishing attacks can be particularly effective, as victims are more likely to lower their guard and engage with the malicious content.

E-mail masquerade attacks are among the most common and most well-studied (Khonji et al., 2013) digital vectors for phishing attacks. Language generation has been studied in correlation to them even before GPT-3 came to the forefront, and has been shown to be effective at creating a variety of attacking strategies with only few hours of effort (Baki et al.,

2017). Aside from providing an important method to scale up email-based attacks, language generation also renders traditional spam and phishing filters less effective.

One of the most widely used method for distinguishing legitimate emails from spam is Bayesian filtering. This approach works by analyzing each word in an email and comparing it against a database of words commonly found in spam messages. The filter calculates the probability of the email being spam based on the presence and frequency of these words. If the likelihood exceeds a certain threshold, the email is flagged as spam. Despite some shortcomings, such as being vulnerable to adversarial manipulation of the underlying frequency tables, Bayesian filtering remains an effective tool for detecting and blocking unwanted emails.

NLG systems could throw a wrench in this common approach, since large language models are capable of higher and less predictable lexical variety than traditional NLG systems. Community Targeted Phishing, as described by Giaretta and Dragoni (2019) aims explicitly to analyze the language of specific groups to craft complex emails, which with the help of today's LLMs can more easily evade traditional defenses. For example, a conversational LM could be trained on the style and publications of a well-known researcher, to then be used as a phishing actor against colleagues in the same area, who may not be directly familiar with the renowned figure. This clone may much more convincingly get victims to surrender personal information and click on malicious links, while evading.

The more dangerous variety of phishing attacks, however, is *spear phishing*. Spear phishing focuses on a single individual, using personalized information gathered from research to craft a highly convincing and tailored attack. This makes spear phishing more difficult to detect, as the attacker often impersonates someone the target knows or trusts, increasing the likelihood of success. The personalized nature of spear phishing often leads to greater harm, as the attacker aims to extract highly sensitive information or gain unauthorized access to critical systems.

The bigger limitation of these attacks – from the perpetrator's perspective – is that much time investment and human time is needed for each target. NLG systems could provide a tool for scaling such attacks, for example by being employed in early stages for automated conversations, thus gaining the victim's trust before the attack proper.

The language generation features provided by modern LLMs thus pose a substantial threat when it comes to their ability to boost the effectiveness of phishing attacks. Their ability is well suited to tricking large groups of people indiscriminately, as well as to strategies that prove insidious to threaten specific communities. They also enable the scaling up of phishing attacks targeted at individuals, since they can engage in conversation, preparing the field for the malicious extraction of sensitive information. The industry's tools for protection against traditional attacks are rendered somewhat obsolete when LLMs enter the picture, thus effectively detecting when a language model is being deployed is crucial.

## 3.2 Information poisoning and influence campaigns

Another area where NLG can cause a negative impact is information and communication. Disinformation campaigns have been extremely common across all of human history. One coon look as back as the Donation of Constantine, a false document used in the 13th century to justify the territorial claims of the papal state, or as recently as the election meddling by foreign states in the US elections of the 21st century.

Disinformation campaigns in particular have traditionally suffered from being dependent on human editors to compose the message and select the diffusion vector – be it emails, social media, blogs, and so on - therefore performing them at scale comes at a steep cost.

Language models have the potential of solving alleviating this limitation. By relying on NLG to generate most of the content of the campaign messages, it is possible for attackers to forgo – at least in part – this resource and time-consuming factor.

Buchanan et al. (2021) explore the role of GPT-3 in a human-machine setting, studying therefore the effectiveness of GPT-3 in generating disinformation campaigns when paired with human-crafted prompts along the process. Their findings suggest that while GPT-3 is unlikely to completely replace human operators in disinformation efforts, it significantly enhances their capabilities by enabling the creation of moderate- to high-quality content at an unprecedented scale. This amplification of output could make disinformation campaigns more pervasive and harder to counter, highlighting the need for robust strategies to mitigate such threats.

Among the various activities that have been identified in disinformation campaigns, GPT-3 was observed to perform exceptionally in some, while needing stricter human supervision in others. It excelled in *narrative reiteration*, a simple task that involves generating multiple variations of a short message that repeat a particular theme. For example, when providing the model with a number of successful climate-change denier tweets, researchers were able to obtain high-quality approximations on the first try and without particular refinery. With how low the effort to mass-produce this type messaging seems to be compared to the effectiveness of the output, it stands to reason that LLMs could enable a previously unseen proliferation of deceitful content.

If reiteration focused on generating tweet-length documents, *narrative elaboration* expands this to medium to long length content, such as articles and blog posts. For this task, GPT-3 was tested for its ability to sound reputable in its output, i.e. in its ability to replicate the tone of a renowned news source – unfortunately the believability of the generated articles was not verified in this setting. By the same token, this task certainly requires more human supervision, whose effects and variability are hard to design an experiment around, than simple reiteration, since getting the prompt right will influence the tone and message of the generation immensely (for example, the headline or sample text to emulate must provide enough information to the model to recognize the style and worldview). Nevertheless, GPT-3 was able to fool classifiers only partially: the evaluation model, which classified genuine articles with over 90% accuracy over three news networks, had only around 65% accuracy. This indicates that GPT-3 is good at the elaboration task, but not perfect. It should however be said that fine-tuning the model could make it more potent in this regard – a fact that was observed with GPT-2 but could not be done on GPT-3 due to technical limitations.

*Narrative seeding* sets itself apart from the previous tasks, in that it requires that the model come up with original conspiratorial material. The objective in this case would be akin to the rise of QAnon between 2017 and 2020, a Twitter account that would post short messages containing conspiracy theories. While the effectiveness of most disinformation campaigns is, at least in part, a function of the scale of the operation, in this case a single human can be an effective catalyst, as long as the content is captivating. While GPT-3 appears to have the capability to come up with such messages – and its proneness to hallucinations might even help it do so – it is not obvious how one would go about measuring its potential in this endeavor.

Exploiting existing stories and divisions that exist in the target societies and groups is equally as important in disinformation campaigns as fabricating importance. *Narrative wedging* involves locating a source of conflict and widening it with disinformation. In the original experiment, messages were generated to influence US religious groups (Christian, Arabic, Jewish) to vote a certain way (Democrat, Republican, Abstain). The process by

which the messages were obtained relied very heavily on human-machine cooperation, where a team of humans would identify the best arguments for a target group to vote a certain way through repeated prompts, and the best arguments were then used for message generation. The resulting 110 messages were reviewed by four experts, with 95% of messages found credible by at least one, and 65% by all four. Surprisingly, some of the messages were quite insidious, with the model soliciting Arabic voters (a traditionally left-leaning block) to vote Republican by calling for them to vote based on individual interests rather than group affiliation.

A similar task, *narrative manipulation*, involved identifying existing news stories and rewriting them to fit a particular narrative. Researchers used GPT-3 to rewrite a series of news articles concerning US political events around 2020, such the Black Lives Matter protests and the incipit of Covid-19. Again, this involved thick interplay between a human team and GPT-3, as the rewriting process was found to be most effective when it was executed in a series of steps (in short: summarize the article, alter the summary, expand again into longer text). Articles were modified to reflect both a left and a right-leaning position, and were then presented to human evaluators to test the effectiveness of the alteration, along with the corresponding authentic reference articles for comparison. This was proven to be mostly the case, i.e. GPT-3 was shown to be consistently able to not only give articles the correct spin, but to even output content that was found to be mostly authentic by the reviewers (the original articles received an average authenticity score of 3.8 on average, whereas generated articles only 2.4, but the best generations were believably authentic).

Finally, *narrative persuasion* tested GPT-3's performance in creating tailored messages for specific targets in an attempt to alter their convictions. Messages were generated for two topics (US involvement in Afghanistan and China), both for and against each issue, with the target's party affiliation in mind. Of 20 generations in each pairing, the 10 best were selected by a human team, and presented to over one thousand human respondents. The participants generally found GPT-3's statements to be convincing, with 63 percent rating them as at least somewhat persuasive, even when the statements were targeted at an audience with opposing political views. While only about 12 percent found the statements "extremely convincing," the majority still found them somewhat persuasive. Additionally, the survey indicated that GPT-3's statements were effective in shifting respondents' opinions on various topics, with a notable and sometimes stark increase in support for the positions advocated by GPT-3 after being exposed to its arguments (in regards to action against China, GPT-3 managed to overturn a situation of absolute majority in favor of sanctions by reducing the pro-sanction faction from 51% to 33% of respondents).

In summary, GPT-3 performed particularly well with *narrative reiteration* (e.g. generating a series of tweets based on an original) and *narrative elaboration* (e.g. generating a blog post or article based on a title or paragraph) – which consist in the generation of short or medium length messages given a particular world view and a prompt. When malicious actors have a well-defined message that they seek to advance, NLG systems seem therefore to be a dangerous tool at their disposal, as they performs exceptionally well in this setting, without requiring particularly skilled operators. GPT-3 was observed to perform worse – or to require more careful prompting from the operator – for tasks requiring higher levels of creativity and originality. *Narrative wedging*, for example, aims to identify a source of division in the target society or group, and amplify its effects by means of carefully crafted disinformation messages. Similarly, *narrative manipulation* is a technique that involves framing existing emergent stories to align with a pre-defined world view. Neither task requires coming up with outright lies, and for both a human-machine team was observed to be necessary to

obtain credible and convincing messages.

Buchanan et al. (2021) go on to highlight that GPT-3's writing characteristics – which can likely be extended to several other language models – suits the goal of disinforming more than that of informing. Indeed, GPT-3 is known to frequently go on tangents and make information up even in neutral to benign generation contexts. These traits might unfortunately make GPT-3 more effective in creating misleading or false narratives than in producing reliable, accurate content, underscoring its potential risks in the hands of those seeking to deceive.

Generic and task-specific language models can also be employed to exploit online review systems to influence consumer behavior. LLMs can be used to generate fake reviews with a target sentiment based on a prompt or a base review to emulate. This is an evolution of the *crowdturfing* method, which involves paying large numbers of people to write fake reviews. Due to the associated economic cost, these attacks were limited to large-scale attacks, and therefore carried a built-in limiting element.

Adelani et al. (2019) devise a strategy that employs language models to compose fake reviews based on an example displaying the desired sentiment towards the thing being reviewed. The generated fakes do not undergo manual review later, but employ a BERT-based sentiment classifier to filter out counterproductive items. To take it a step further, they use a fine-tuned version GPT-2 and BERT on Amazon and Yelp review databases, an operation that perhaps wouldn't necessarily be typical of attacks based on GPT-3 and other modern LLM variants. Nonetheless, using a task-specific version of GPT-2 might make it more comparable to task-agnostic, modern LLMs such as GPT-3 and GPT-4, making the findings more generalizable to modern systems.

To verify the fooling power of the fake reviews, sets of 3 fake reviews and 1 real review were presented to human evaluators, with the objective of correctly finding the "real" review. The results demonstrate that participants often struggled to discern the authentic review, as their success rates were close to the random chance rate of 25%. This suggests that the evaluators were essentially guessing, indicating that the fake reviews were convincing enough to be mistaken for real ones. Such findings indicate that even in the premodern era of language modeling (i.e. before GPT-3), NLG already proved a valid substitute for the more expensive crowdturfing attacks.

## 3.3 Faking human authorship

Other malicious uses of LLMs rely on its ability to appear human, and not so much on its ability to specifically compose high-quality text. One such application is the generation of pseudo-scientific papers to contribute to one's research numbers. Examples of generation of such papers are common even before language modelling became popular. SCIgen (Hargrave, 2005) is a tool introduced in 2005 that allowed for the generation of nonsensical scientific papers by using a context-free grammar based generator. Such papers even pass the peer-review process from time to time and appear in respected publications. As recently as 2021, Cabanac and Labbé (2021) identified 197 SCIgen articles for which there had been no withdrawal notice, and were at times even being sold.

While SCIgen can be identified with relatively low computational effort and high accuracy, detecting content produced by more advanced generation tools, such as LLMs, presents a far greater challenge. With their ability to produce highly convincing and coherent text, these models could significantly worsen the issue, particularly in non-peer-reviewed technical documents. Beyond merely disrupting the scientific process, the proliferation of AI-generated technical content could also be exploited to influence public opinion and create confusion

around important scientific issues, making it a growing threat in both academic and public discourse.

(Un)fortunately, there don't appear to be studies investigating the concrete prevalence of LLM-generated scientific papers across the recent academic landscape – perhaps both due to the recency of the phenomenon, as well as the difficulty in detecting the employment of modern NLG systems. Rodriguez et al. (2022) investigated possible detection strategies with self-made document tampering, but used the older GPT-2 for document contamination as opposed to more modern versions, therefore their results are hard to generalize (admittedly, this example investigate domain-specific fine-tuned generation, something that is harder and more expensive to do with future versions of GPT). Nevertheless, this study provides valuable insight into the future of LLM-generated papers. Further discussion of their detection strategies can be found in Chapter 4.

Among the overtly malicious uses of NLG, there are other that do not seek to harm directly. Students using language models to, either partially or completely, complete their coursework for them is one such case. Overreliance on systems like ChatGPT can discourage students from developing analytical and thinking skills. Passively incorporating the generated answers is a real danger when interfacing with such systems without an adequate evaluation framework – checking for factual correctness is one element among many that need checking before accepting a generation.

Another indirectly harmful application of NLG is content generation for social media. AI-generated text and images risk diluting the true human experience present on the platforms, to such an extent that they may become inhospitable places to organic users.

Furthermore, anyone using LLMs, however well-intentioned, is exposing themselves to the risk of plagiarism. The importance of checking the model output before incorporating it – especially in formal and academic writing – needs hardly be restated. Gao et al. (2022) investigated abstract generation for papers in the biomedical domain, and found no plagiarism in the output, though it did observe high accuracy in predicting whether an abstract was generated or human-written. This finding alleviates concerns somewhat, though the risk of inadvertent plagiarism is never fully 0, especially in longer contexts.

## References

D. I. Adelani, H. Mai, F. Fang, H. H. Nguyen, J. Yamagishi, and I. Echizen. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection, 2019. URL https://arxiv.org/abs/1907.09177.

J. A. Anderson. *An introduction to neural networks*. MIT press, 1995.

I. Asimov. *Foundation*. Foundation series. Gnome Press, 1951. URL https://books.google.it/books?id=gFpQswEACAAJ.

S. Baki, R. Verma, A. Mukherjee, and O. Gnawali. Scaling and effectiveness of email masquerade attacks: Exploiting natural language generation. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, page 469–482, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349444. doi: 10.1145/3052973.3053037. URL https://doi.org/10.1145/3052973.3053037.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.

B. Buchanan, A. Lohn, M. Musser, and K. Sedova. Truth, lies, and automation. *Center for Security and Emerging technology*, 1(1):2, 2021.

G. Cabanac and C. Labbé. Prevalence of nonsensical algorithmically generated papers in the scientific literature. *Journal of the Association for Information Science and Technology*, 72 (12):1461–1476, 2021.

M. Chen et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394, 1999.

E. Crothers, N. Japkowicz, and H. Viktor. Machine generated text: A comprehensive survey of threat models and detection methods, 2023. URL https://arxiv.org/abs/2210.07321.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

G. A. Fowler. Say what, Bard? What Google's new AI gets right, wrong and weird, 2023. URL https://www.washingtonpost.com/technology/2023/03/21/google-bard/.

C. A. Gao, F. M. Howard, N. S. Markov, E. C. Dyer, S. Ramesh, Y. Luo, and A. T. Pearson. Comparing scientific abstracts generated by chatgpt to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. *BioRxiv*, pages 2022–12, 2022.

J. Gao and C.-Y. Lin. Introduction to the special issue on statistical language modeling, 2004.

A. Giaretta and N. Dragoni. *Community Targeted Phishing: A Middle Ground Between Massive and Spear Phishing Through Natural Language Generation*, page 86–93. Springer International Publishing, Mar. 2019. ISBN 9783030146870. doi: 10.1007/978-3-030-14687-0_8. URL http://dx.doi.org/10.1007/978-3-030-14687-0_8.

J. Hargrave. SCIgen - An Automatic CS Paper Generator, 2005. URL https://pdos.csail.mit.edu/archive/scigen/.

A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b, 2023.

S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3): 400–401, 1987.

M. Khonji, Y. Iraqi, and A. Jones. Phishing detection: A literature survey. *IEEE Communications Surveys & Tutorials*, 15:2091–2121, 2013. URL https://api.semanticscholar.org/CorpusID:4133482.

C.-C. Lin, A. Jaech, X. Li, M. R. Gormley, and J. Eisner. Limitations of autoregressive models and their alternatives, 2021. URL https://arxiv.org/abs/2010.11939.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/abs/1301.3781.

OpenAI et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

J. D. Rodriguez, T. Hay, D. Gros, Z. Shamsi, and R. Srinivasan. Cross-domain detection of gpt-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: human language technologies*, pages 1213–1233, 2022.

S. Selva Birunda and R. Kanniga Devi. A review on word embedding techniques for text classification. *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020*, pages 267–281, 2021.

A. Shostack. *Threat Modeling: Designing for Security*. Wiley Publishing, 1st edition, 2014. ISBN 1118809998. URL https://www.wiley.com/en-us/Threat+Modeling%3A+Designing+for+Security-p-9781118809990.

G. Team et al. Gemini: A family of highly capable multimodal models, 2024. URL https://arxiv.org/abs/2312.11805.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. LLaMA: Open and efficient foundation language models, 2023.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

J. Weizenbaum. ELIZA — a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

J. Zhang. Gradient descent based optimization algorithms for deep learning models training, 2019. URL https://arxiv.org/abs/1903.03614.

## Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Masterarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum Unterschrift