Università
Ca'Foscari
Venezia

Master's Degree Programme

in Language Sciences

Final Thesis

Syntactic and Emotional Properties of AI-Generated Texts: a Corpus-Based Comparison

Supervisor

Ch. Dr. Gianluca Lebani

Assistant supervisor

Ch. Dr. Alice Suozzi

Graduand

Veronika Gersh

901362

Academic Year

2025 / 2026

Table of Contents:

Abstract

This thesis examines the linguistic properties of texts generated by Large Language Models in comparison with human-authored corpora. It focuses on three core questions: the extent to which AI-generated corpora approximate human texts in syntactic complexity, the influence of prompt complexity on output characteristics, and the distribution of sentiment and emotion in machine-produced language. Three corpora were created using the OpenAI API (version o1) under systematically varied prompting conditions and compared with three human reference corpora: OpenSubtitles, a Children's Stories Text Corpus, and the Leipzig Web Corpus. The analysis combined established syntactic complexity metrics and dependency-based measures of structural depth with sentiment modelling through a pre-trained emotion detection model built on the RoBERTa architecture (i.e. Emotion English DistilRoBERTa-base).

The findings show that AI-generated texts display grammatical fluency, but exhibit lower syntactic variety and greater structural regularity than human corpora, resembling the simplicity of subtitles rather than the richness of children's literature or web texts. Dependency-based measures further revealed a preference for efficient, low-cost constructions. Prompt complexity was found to shape outputs significantly, with more elaborate prompts eliciting greater syntactic diversity, though never fully matching human-like range. Sentiment analysis indicated a strong bias toward neutral and mildly positive affect, with limited representation of negative emotions.

The study contributes to computational linguistics by offering a detailed, corpus-based comparison of human and AI texts, highlighting the methodological role of prompt engineering, and underlining both the promise and the constraints of Large Language Model outputs. It concludes that while prompt design can narrow the gap between human and machine-authored texts, AI-generated corpora remain distinguishable in their syntactic regularities and emotional limitations, making them valuable but imperfect substitutes for authentic human language.

Introduction

The emergence of *Large Language Models* (LLMs) appears to be one of the most significant developments in the spheres of computer science, particularly in computational linguistics and natural language processing. Their application varies from automated machine translation tools to more complex ones like customer service, educational technologies or scientific research (Sabry et al., 2024; Dulaney et al., 2023). While these strategies keep on developing, they can face some challenges, such as hallucinations of the models (Maynez et al., 2020) or their biased gender representations (Jana et al., 2024).

The main idea of this thesis is centred on the issue of whether LLMs can produce text that can be similar to human-written text in terms of fluency and grammaticality. Pieces of text written by Generative AI can sometimes be considered as simplified in clause structure, syntactic variety or figurative expressions (Liu et al., 2022), which can cause potential problems in automatic language generation without the integration of human-based verification, especially in fields like education or medicine.

A further motivating factor is the increasing prominence of prompt engineering. LLMs do not operate autonomously; they respond to user input, and the structure, wording, and complexity of prompts significantly influence their outputs. The application of different prompting levels and strategies can potentially help reduce bias or hallucinations in outputs (Lin, 2024). This study addresses the gap in the research concerning prompt complexity through a corpus-based comparison of AI-generated and human-authored texts. The research focuses on three questions:

1. **How can AI-generated corpora be compared to human-authored corpora in terms of syntactic complexity?**

This includes an analysis of the syntactic complexity metrics, based on which the syntactic richness, the distribution of the syntactic structures and syntactic density can be tested.

2. **What role does prompt complexity play in shaping the linguistic characteristics of AI-generated corpora?**

The application of different prompting levels can help to investigate the extent to which the complexity of the prompt can influence the output of the model and assist in approximation to human-like language.

3. **How do AI-generated texts differ from human corpora in their expression of sentiment and emotion?**

Beyond syntax, this research also introduces the examination of emotional distribution, assessing whether AI texts change this distribution based on the prompt complexity.

The methodology of the study includes: the generation of the three corpora via the OpenAI API (version o1) (OpenAI, 2025d), each based on 15 prompts constructed in line with the LOB corpus sampling frame (Hofland & Johansson, 1982). Prompt complexity was systematically varied across groups, ranging from minimal instructions to detailed, structured queries. The generated corpora were compared with three human-authored reference corpora: OpenSubtitles English Monolingual Data (Lison & Tiedemann, 2016), Children Stories Text Corpus (that presents cleaned Gutenberg books) (Eden, 2021) and web-crawled Leipzig Web Corpus (Goldhahn et al., 2012).

Quantitative analysis focused on six syntactic complexity metrics widely used in computational linguistics: Mean Length of Clause (MLC), Mean Length of Sentence (MLS), Clauses per Sentence (C/S), Dependent Clauses per Clause (DC/C), Coordinate Phrases per Clause (CP/C), Complex Nominals per Clause (CN/C) (Kyle & Crossley, 2018; Kyle et al. 2021; Liu & Afzaal, 2021; Liu et al., 2023) In addition, dependency-based metrics capturing integration costs and locality principles were applied to assess structural depth and processing load: Incomplete Dependency Theory Metric (IDT), Dependency Locality Theory Metric (DLT), Combined IDT+DLT Metric (IDT+DLT), Nested Nouns Distance Metric (NND), Left-Embeddedness Metric (LE) (Zou et al., 2022; Rathi, 2021; Mirzapour et al., 2018). Sentiment and emotional analysis were conducted using the Emotion English DistilRoBERTa-base model (Hartmann, 2022).

The significance of this thesis lies in its contributions to both theoretical and practical aspects of computational linguistics research. From a theoretical perspective, the study advances understanding of how LLMs approximate human linguistic behaviour. From a methodological perspective, it demonstrates how corpus-based tools can be adapted to evaluate AI outputs, offering a replicable framework for future research.

The structure of the thesis reflects these aims. Chapter 1 provides an overview of text generation in computational linguistics, including the development of LLMs. Chapter 2 introduces prompt engineering, outlining key strategies and their linguistic implications.

Chapter 3 reviews lexical, syntactic, discourse, and stylistic features of AI-generated texts, as well as challenges in detection. Chapter 4 presents methodological approaches to corpus comparison. Chapters 5–7 describe the design, data collection, and analytic procedures of the present study. Chapter 8 reports results from syntactic analysis, while Chapter 9 expands on and discusses the findings. Chapter 10 discusses the challenges and future directions of the study.

By examining how prompt complexity shapes AI-generated corpora and comparing them systematically with human-authored texts, this thesis argues that while LLMs can approximate human-like text under carefully designed conditions, they remain distinguishable in their syntactic regularities and emotional distributions. These insights are vital for linguists, educators, and computational researchers navigating the evolving relationship between human and machine language.

Chapter 1: Large Language Models and Text Generation

1.1 Introduction to Text Generation

Text generation “aims to produce plausible and readable text in a human language, from the input data in various forms including text, image, table and knowledge base” (Li et al., 2024a) using *Natural Language Processing* (NLP) techniques, which can be presented in various forms, such as sentences, paragraphs, or full documents; the main goal of this technique is not only to produce grammatically correct utterances but also to make them indistinguishable from texts produced by humans. This approach has practical importance in question answering, machine translation, text summarisation, grammar correction, story generation, and conversational dialogue (Jurafsky & Martin, 2024). Text generation, image generation, and code generation became a part of a growing field within artificial intelligence commonly referred to as *generative AI* (Jurafsky & Martin, 2024). This term refers to the “computational techniques that are capable of generating seemingly new, meaningful content such as text, images, or audio from training data” (Feuerriegel et al., 2024).

Text generation can be applied to various tasks, from natural language understanding to more specialised applications, such as drug discovery, mentioned by Wang et al. (2024).

Text-generation models are pivotal in Natural Language Generation (NLG); they generate coherent text, making them valuable for content creation, including automated writing in journalism, marketing, and entertainment (Jana et al., 2024).

Text-generation models are the backbone of Conversational Agents and Chatbots, providing efficient customer support, information retrieval, and task automation. These models help chatbots engage in meaningful dialogues, improving user interactions across various sectors, such as finance, where they reduce reliance on human agents (Jana et al., 2024; Wang et al., 2024).

In content creation, text-generation models automate the generation of articles, blog posts, social media posts or advertisements, and enable businesses to engage their audience more effectively while maintaining high-quality content (Jana et al., 2024).

These models have also revolutionised Machine Translation, enabling accurate translations between languages, though sometimes models are still not as accurate as human translators.

This is crucial for global businesses and international organisations in breaking down language barriers and ensuring contextually accurate translations (Jana et al., 2024).

In education, text-generation models provide personalised learning experiences, offering real-time feedback and fostering student engagement (Wang et al., 2024).

This chapter presents an overview of the benefits and challenges in text generation, exploring basic concepts of the topic; the aim is to establish a clear conceptual and methodological foundation for the subsequent chapters that focus on corpus-based analyses of AI-generated text using various prompting techniques.

1.2 Approaches to Text Generation

The main approaches to text generation in NLP include: *rule-based*, *template-based*, and *machine learning-based* approaches (Van Deemter, 2005; Wells, 2025).

1.2.1 Rule-Based Approaches

In rule-based text generation, the output is produced by applying a set of pre-established patterns that are created by experts for a specific task; examples of such systems include chatbots that use decision trees or content generation systems with predefined templates for text production (Bauer et al., 2015). For instance, early chatbots such as ELIZA (Weizenbaum, 1966) followed fixed if–then rules, responding to input sentences by matching them against predefined patterns (e.g., transforming “I feel sad” into “Why do you feel sad?”). The strength of this method lies in its predictability and interpretability: developers know exactly why the system produces a given output. However, the rules do not adapt to unexpected input, and maintaining them requires ongoing manual updates as user needs or contexts evolve (Shawar & Atwell, 2007).

1.2.2 Template-Based Approaches

Template-based approaches also use predefined patterns, like rule-based approaches, but they are more tailored to the input, offering the possibility to fill these templates with specific words or phrases (Goyal et al., 2023). Examples of template-based systems are content generation systems that automatically produce content by selecting suitable sentence structures from a predefined list to generate text based on input data (Van Deemter, 2005). For example, a weather-reporting system might use a template to generate textual weather forecasts from representations of graphical weather maps, or produce answers to questions about an object described in a knowledge base (Reiter & Dale, 1997). This allows for more variation than pure rules, since a single template can produce many outputs. Compared with rule-based systems, template-based approaches provide greater adaptability and efficiency, but the creative range of outputs is still limited by the number and diversity of the templates designed. Unlike machine learning–based approaches, they cannot easily generate novel phrasing beyond what the templates allow (Van Deemter, 2005).

1.2.3 Machine Learning-Based Approaches

Machine learning-based approaches typically utilise *Recurrent Neural Networks (RNNs)* or *transformer models* to capture the patterns and dependencies within a given text corpus (Huang & Yu, 2022).

RNNs are a type of neural architecture characterised by their feedback loops, based on which they can remember previous information from the inputs in their memory and answer questions based on context (Das et al., 2023). They are used mainly in the areas that are connected to making predictions about future outcomes, such as stock market predictions or sales forecasting (Bao et al., 2025; Hu et al., 2021; Hewamalage et al., 2020)

The transformer architecture, a type of neural architecture first mentioned by Vaswani et al. in the paper “Attention is All You Need” (Vaswani et al., 2017), has become the basic architecture for the construction of LLMs (Jana et al., 2024). Unlike RNNs, transformers don’t process words one at a time; they capture the entire sentence using a self-attention mechanism to identify dependencies between words. An *attention mechanism* is a technique used in machine learning that allows models to focus on the most important parts of the input data and ignore minor details that are not crucial at the moment (Bergmann & Stryker, 2024). From a mathematical perspective, an attention mechanism calculates attention weights, which indicate the importance of each part of an input sequence for a given task (Bergmann & Stryker, 2024). Attention models are trained to accurately assign these weights through supervised or self-supervised learning on large datasets. This mechanism makes models very powerful and accurate, especially for long sentences, though they require a lot of memory and computing power (Jana et al., 2024). The most well-known examples of LLMs working on transformer architecture are OpenAI's GPT (Generative Pretrained Transformer) series (Brown et al., 2020) and Google's BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019).

Examples of machine learning-based systems are language translation systems that use neural networks to translate texts from one language to another, or content generation systems that use neural networks to generate novel output based on input data (Tan et al., 2020). The advantage of machine learning-based systems is their creativity and novelty in the generated output, and they are more accurate compared to rule-based or template-based systems; however, they require training and large amounts of resources to perform (Franceschelli & Musolesi, 2024).

1.2.3.1 Large Language Models (LLMs)

The appearance of LLMs was possible after the introduction of transformer architecture and the application of the attention mechanism to NLP tasks. The adoption of LLMs for text generation has significantly expanded the capabilities of NLP, now encompassing not only natural language tasks but also code generation and image generation, becoming generative AI (Jurafsky & Martin, 2024).

The generation in the LLMs is based on the process of selecting the next word based on contextual probability counted by the model; this prediction process was named decoding (Jurafsky & Martin, 2024). A common decoding technique is called sampling, and it is connected with the idea of a random selection of the successive words in accordance with the probabilities calculated by the model (Jurafsky & Martin, 2024). Words that the model considers more probable in a given context are selected more often, resulting in generated text that closely reflects the model's interpretation of what fits best in the specific context.

According to Jurafsky and Martin (2024), the effectiveness of large language models is primarily influenced by three key factors: "model size (the number of parameters not counting embeddings), dataset size (the amount of training data), and the amount of compute used for training" (Jurafsky & Martin, 2024).

The enhancement of the model performance can be achieved by "adding parameters (adding more layers or having wider contexts or both), by training on more data, or by training for more iterations" (Jurafsky & Martin, 2024); these relationships between the elements mentioned above are known as scaling laws.

1.2.3.2 Training of LLMs

Training of LLMs consists of three stages, according to Liu et al. (2024): data collection, pre-training process and fine-tuning (Liu et al., 2024).

LLMs are mainly trained on texts from the web, since they contain enough information for the model to grasp the patterns of natural human speech production in various forms, so that the model can become suitable for application to various NLP tasks, such as question answering or translations (Jurafsky & Martin, 2024). The most important step on the first stage is that the data borrowed from the web should be filtered for both quality and safety, using the websites that are reliable and avoiding those that contain lots of PII (Personal Identifiable Information) or adult content (Jurafsky & Martin, 2024).

Pre-training is a process of "learning knowledge about language and the world from vast amounts of text" (Jurafsky & Martin, 2024). Training can be performed using self-supervised learning strategies, when a model learns to predict certain properties of the input data without explicit supervision: a corpus of text is taken as training material and at each time step, the model is asked to predict the upcoming word (Jurafsky & Martin, 2024).

During the fine-tuning stage, the supervised learning strategies are usually applied: they involve training a model on labelled data, where the input-output pairs are explicitly provided during training (Liu et al., 2024).

One more stage that has a high level of importance is the final evaluation of the model, when the process of assessing the performance and capabilities of the model takes place (McGrath & Jonker, 2024). Evaluation can be performed using both automated benchmarks and human assessments to identify the strengths and weaknesses of a model: this process includes comparing the model's outputs with ground truth data (information considered accurate) or human-generated responses, to assess the model's accuracy, coherence, and reliability (McGrath & Jonker, 2024). The findings from LLM evaluations assist researchers and developers in the identification of weak points of the model that require enhancement.

1.3 Advantages and Challenges of Text Generation

1.3.1 Advantages of Text Generation

Text generation technologies have introduced considerable advantages across a variety of sectors:

Enhanced Efficiency and Creativity

Text generation technologies provide significant efficiency improvements across multiple sectors. According to Doshi & Hauser (2024), generative AI allows for the rapid production of large volumes of text that would be highly time-consuming for humans to create manually. This includes automated generation of product descriptions, technical documentation, social media content, and other text-heavy tasks. By reducing the time and effort required for these repetitive tasks, human teams can focus on more strategic and creative activities. Additionally, AI can act as a creative collaborator, generating novel ideas, alternative phrasings, or perspectives that human teams may not have considered. This capability has been shown to enhance overall creativity, particularly for individuals or teams who might struggle with generating content under strict time constraints (Doshi & Hauser, 2024).

Increased Accessibility

Another major advantage of AI text generation lies in its capacity to improve accessibility. AI-generated content can be adapted to serve individuals with disabilities or language barriers, thereby promoting inclusivity. For instance, Shen et al. (2025) introduced AltGen, an AI framework designed to automatically generate alternative text for EPUB files, facilitating access for visually impaired readers. Similarly, AI-based translations, summarisations, and content simplifications enable non-native speakers and users with diverse learning needs to access and comprehend information that might otherwise be difficult to obtain (Shen et al., 2025). By providing content in multiple formats and languages, these systems reduce barriers to information and knowledge.

Better Customer Engagement

Personalised content generation is another key benefit of AI. AI-driven systems allow companies to create content tailored to individual users' preferences, increasing engagement and satisfaction. According to Bansal et al. (2025), AI tools can analyse user data to provide more targeted communications, recommendations, and marketing messages, thereby improving interactions between businesses and their customers. This capability not only helps companies reach a wider audience but also fosters stronger connections with individual users through personalised experiences (Bansal et al., 2025).

Enhanced Language Learning

AI text generation also offers significant potential in educational contexts, particularly for language learning. Habib et al. (2024) found that generative AI tools can support learners by providing immediate, personalised feedback on writing tasks and offering examples of different linguistic styles and genres. This facilitates a structured approach to skill development, allowing learners to practice writing, comprehension, and creativity in a controlled environment. By simulating realistic language use, AI can complement traditional language learning methods and adapt to learners' individual needs (Habib et al., 2024).

1.3.2 Challenges of Text Generation

Despite the above-mentioned advantages, text generation systems can encounter multiple challenges during their application process:

Quality and Coherence

Factual reliability of the generated text remains problematic, since machine learning-based text-generation systems are prone to producing hallucinations (Maynez et al., 2020). The term *hallucinations* refers to a situation when generative AI produces content that is "nonsensical or unfaithful to the provided source content" (Ji et al., 2024).

Though these models are trained specifically to generate text that is predictable and coherent, the training algorithms usually don't possess a strategy to test the generated text for correctness and truthfulness (Jurafsky & Martin, 2024). It can be partially corrected using particular prompting techniques, and these approaches will be discussed in the next chapters. This limitation is especially critical in sensitive domains like healthcare or law.

Since text generation systems are trained on large collections of textual information available on the web, including social media, it is not always possible to control all probable biases presented there; this leads to potentially harmful responses generated by the models. They were noticed to exhibit gender biases, which, according to Jana et al. (2024), may not only result in biased language generation but also affect "applications such as job recruiting, personal assistants, and content recommendation systems" (Jana et al., 2024).

Just as with gender bias, text generation models can also reflect racial biases embedded within their training datasets; this can result in unfair or discriminatory impacts in fields like criminal justice, recruitment, and translation services (Jana et al., 2024).

Copyright, Data Consent and Privacy Concerns

A significant part of the information used to train text-generation models, like books or even websites, is protected by copyright. In some countries, like in the United States, according to the law, it is not prohibited to use copyrighted information for transformative uses, while in others, these rules are not applicable (Jurafsky & Martin, 2024).

To deal with the privacy issues, website owners have mechanisms to signal that their content should not be accessed by web crawlers, either through the use of a robots.txt file or stated terms of service (Jurafsky & Martin, 2024). As it was stated by Longpre et al. (2024a, as cited in Jurafsky & Martin, 2024), in recent times, there has been a notable rise in the number of websites expressing objections to having their content used in the training of text-generation models. This problem may have a significant impact on the production of the web corpora and training of the models, since it reduces the information available for training.

Moreover, datasets gathered from the web often raise privacy concerns, as they may include sensitive data such as personal contact details, legal documents, client communications, and proprietary data, making them attractive targets for cyberattacks or data theft (Gstrein & Beaulieu, 2022). Although filtering techniques are employed to screen out sources that are likely to contain significant amounts of private information, these methods do not offer complete protection.

This problem leads to another issue of misuse of the model possibilities, creating deepfakes: text-generation models have the potential to generate text, audio and even videos that contain false information, creating fake news articles, social media posts, and reviews, damaging the credibility of information sources (Abbas & Taeihagh, 2024).

Chapter 2: Theoretical Background on Prompt Engineering

2.1 Defining Prompt Engineering

According to Vatsal & Dubey (2024), *prompt engineering* is "the process of creating natural language instructions, or prompts, to extract knowledge from LLMs in an organized manner". In this context, the term *prompt* can be understood in two ways: it can signify the linguistic input one writes down to a generative AI tool, such as a simple question or instruction. For example, a user might write: "Tell me what the term *Generative AI* means", which guides the model to produce creative text in response to a clearly defined request. Second, a prompt can be structured with programmatic instructions that influence the model's behaviour more formally, often through embedded rules or API-level parameters (discussed in detail in the next paragraph).

There are certain techniques to work explicitly with the linguistic input to design an efficient prompt, outlined by various guides (). Beyond these text-oriented techniques, a more controlled and versatile method of prompting involves direct interaction via application programming interfaces: *API programming* (Lamothe et al., 2021), which is "the interface to a reusable software entity used by multiple clients outside the developing organization, and that can be distributed separately from environment code" (Robillard et al., 2013). Employing APIs for prompt engineering provides a heightened level of control over how inputs are structured and processed. As van der Vlist et al. (2022) note, it enables the user to structure the prompt carefully and ensures precise control over the input-output dynamics, supporting fine-tuning of responses according to multiple parameters such as length, style, or semantic precision. Moreover, contemporary AI platforms increasingly expose adjustable settings through APIs that facilitate optimisation of model behaviour (OpenAI, 2025d). By combining carefully constructed linguistic prompts with API-driven control, users can achieve a balance between natural language expressivity and computational precision, a synergy that is central to effective prompt engineering and will be examined in greater detail in the following sections.

The significance of prompt engineering lies in its ability to let users experiment and manipulate inputs to influence output variation, as altering the linguistic structure of a prompt can result in markedly different outputs. For the reason that LLMs do not comprehend the text in a way that humans do, and they can only imitate the actual speech, prompting engineering allows users to guide the model's responses more effectively, ensuring that the generated output aligns with their intentions and desired outcomes, minimising errors, biases and hallucinations.

Moreover, prompt engineering can also serve as a tool for creating machine-generated responses for various industries, such as journalism (Cheng, 2025; Sarhan et al., 2025), customer service (Ilagan et al., 2024) or academic research (Dulaney et al., 2023). For instance, businesses rely on prompting to generate content for social media or their web pages (Guriță, 2025); in the academic sphere, specific prompting strategies serve to search for resources (Önder & Akçapınar, 2023); educators use prompts to create lesson plans or quizzes (Sabry et al., 2024; Zhou et al., 2024a). All these examples mentioned show that mastering prompt engineering becomes vital in various spheres of human life, even if it is sometimes not noticeable.

2.2 Types of Prompting Strategies

There are numerous types of prompting strategies, each serving a specific purpose and they can be applied depending on the intended outcome. This chapter will explore only the major types of prompting strategies that are most well-known and widely used, some of which were adopted from the classification suggested by Sahoo et al. (2024) and presented in Table 1, and the rest from other available sources.

Table 1. Prompting strategies from Sahoo et al. (2024)

Strategy Type / Goal	Specific Prompting Approach Example
New Tasks Without Extensive Training	● Zero-shot Prompting ● Few-shot Prompting ● Chain-of-Thought (CoT) Prompting ● Automatic Chain-of-Thought (Auto-CoT)
Reasoning and Logic	● Self-Consistency ● Logical CoT (LogicCoT) ● Chain-of-Symbol (CoS) ● Tree-of-Thoughts (ToT) ● Graph-of-Thought (GoT) ● System 2 Attention Prompting ● Thread of Thought (ThoT) ● Chain of Table Prompting

Reduce Hallucination	● Retrieval Augmented Generation (RAG) ● ReAct Prompting ● Chain-of-Verification (CoVe) ● Chain-of-Note (CoN) ● Chain-of-Knowledge (CoK)
User Interaction	● Active-Prompt
Fine-Tuning and Optimisation	● Automatic Prompt Engineer (APE)
Knowledge-Based Reasoning and Generation	● Automatic Reasoning and Tool-use (ART)
Improving Consistency and Coherence	● Contrastive Chain-of-Thought Prompting (CCoT)
Managing Emotions and Tone	● Emotion Prompting
Code Generation and Execution	● Scratchpad Prompting ● Program of Thoughts (PoT) ● Structured Chain-of-Thought (SCoT) ● Chain of Code (CoC)
Optimisation and Efficiency	● Optimisation by Prompting
Understanding User Intent	● Rephrase and Respond (RaR)
Metacognition and Self-Reflection	● Take a Step Back Prompting

Prompting techniques fall into two broad categories: hard and soft prompting. The first group of strategies uses manually crafted instructions to manipulate the output, such as zero-shot, one-shot of few-shot prompting strategies, while the second one deals with embeddings that are learned based on the provided examples to adapt to the requirements of a specific task (Xu & Sheng, 2024), such as Prefix-Tuning (Li & Liang, 2021) or Prompt-Tuning (Lester et al., 2021).

2.2.1 Hard prompting strategies

Zero-shot Prompting

Zero shot-prompting provides only the instructions to the model without adding any examples to help with the generation process (Li, 2023); this strategy offers several benefits, including high interpretability, minimal need for training data or examples, a simplified design process focused only on task instructions, and a flexible structure that allows input to be placed as required (Li, 2023).

While zero-shot prompting is efficient in some contexts, it may not always guarantee suitable results, since the lack of contextual information can sometimes cause hallucinations or incorrect interpretations; but some scholars argue it stating that well-designed zero-shot prompts can outperform few-shot prompts, as examples are not always perceived by the model as clear guidance (Reynolds & McDonell, 2021).

The example for this strategy can be: “Make the summary of the following article in one paragraph”. There is no description provided of the possible length of the paragraph, or no clear guidance for the model on which specific points should be highlighted, so the user will likely ask for some changes after receiving the output, which is more time-consuming than implementing some more complex prompting strategies on the very first step.

Zero-shot prompting is particularly useful when tasks are simple, training data is unavailable, or rapid output generation is needed, as it requires minimal preparation and allows flexible adaptation across tasks. However, its lack of contextual examples can lead to hallucinations or incorrect outputs, and it may be less reliable for complex or nuanced tasks.

One-shot Prompting

This strategy relies on providing only one example in the input before generating the output, or it uses a single prompt before getting the answer (Chen et al., 2025). For instance, when it is used in sentiment analysis tasks, one example of classification is provided for one category; for instance, “The food was amazing, and the service was exceptional” is marked as a positive review, and in all the other examples, users rely on the model's decision. Although one-shot prompting introduces some level of context, it still may not fully address ambiguity in responses. The one-shot approach provides minimal contextual guidance, which can improve output relevance over zero-shot prompting and is relatively simple to implement.

However, it is sensitive to the quality of the single example provided and may still result in ambiguous outputs. It is most suitable when only one high-quality example is available and the task is moderately complex, offering a balance between simplicity and contextual guidance.

Few-shot Prompting

Unlike the zero-shot or one-shot prompting, a few-shot one requests several (more than one) examples to train the model before generating the answer (Liu et al., 2021), for instance, in the same example for sentiment analysis tasks the model is provided with examples for all the categories, like positive/negative/neutral, not only one like in one-shot strategy. Moreover, providing a few examples can help the model generate responses that match the desired tone, style and structure of the output. Few-shot prompting is advantageous because it improves accuracy and contextual understanding, guides the model toward the desired style, and reduces ambiguity in the outputs. However, it requires multiple high-quality examples, increases the complexity of prompt design, and can be computationally more intensive. This strategy is particularly suitable when multiple clear examples are available and the task requires nuanced understanding, such as sentiment classification with multiple categories or stylistically specific outputs.

Chain-of-Thought (CoT) Prompting

One of the most common strategies of hard prompting is the Chain-of-Thought (CoT) prompting. It demands the input that is decomposed into smaller units (intermediate steps), making the query more understandable for the analysis of the system, enabling it to generate more correct responses (Wei et al., 2023). This technique is particularly useful in domains like mathematics and logic, where reasoning steps need to be made explicit to ensure correctness; for example if we ask ChatGPT to reply on this input: "Mary has 12 apples; she gives 4 to John and then her mom gives her 6 more. How many apples does she have now? Let's think step by step" the model's response will be broken down into several steps:

"Sure! Let's break it down step by step: Mary starts with 12 apples, gives 4 to John (12 - 4 = 8), then receives 6 more from her mom (8 + 6 = 14), so she now has 14 apples" (OpenAI, 2025a).

Tree-of-Thoughts (ToT) Prompting

This approach is built on the Chain-of-Thought method by organising intermediate reasoning steps into a tree-like structure, which enables the system to evaluate the outputs and, based on this, make the decision (Sahoo et al., 2024). For example, a user may ask to make a list of possible travel destinations for holidays, and then evaluate each of them to find the best one, and the model will come up with one city that, according to it, is the most suitable.

Prompt Chaining

It is a strategy that uses a sequence of prompts rather than just one to divide the output into several smaller parts, especially if the task is a complex, multi-stepped one (Wu et al., 2022). For instance, instead of asking about all the information needed in one prompt, one first asks about more general information in the first query, then gets the response and only after that requests more specific information about a certain point:

Prompt 1: Give me a list of 5 popular tourist attractions in Venice

Model Response: “1. St. Mark's Basilica, 2. Doge's Palace, 3. Rialto Bridge, 4. Grand Canal, 5. Murano Island” (OpenAI, 2025b)

Prompt 2: Give me historical information about St. Mark's Basilica

Prompt 3: List any special artworks found inside it

Contextual Prompting

This type of prompting requires providing information about the context of the situation that can be used as background information to make the model understand the specific setting (Muktadir, 2023). When the answer to a question depends on a particular scenario, supplying context can guide the model to generate more accurate or contextually appropriate responses; for instance, if a user needs to write an email specifying that its addressee is a professor at the university may help the model to choose the register and vocabulary properly.

Negative prompting

To avoid certain ideas or things from being mentioned in the output, there is an option to specify all the conditions that should be avoided by the model in the original prompt (Ban et al., 2024). It may help to control the response and ensure that the generated output aligns with the desired guidelines, preventing unintended or inappropriate content; an example of such a prompt may be asking to avoid specific strategies in programming or making a list of stop-words that should not be used in a certain writing.

Meta-prompting

Meta Prompting is an advanced prompting technique whose goal is to provide the structure of the answer to the model, focusing on the format of the future output rather than the context; this strategy can be particularly helpful when understanding the structure or framework of a problem is essential for comprehending or addressing it (Zhang et al., 2023). For example, some types of academic writing require a specific formatting, and providing it before the first output of the model can help structure it in a better way.

Hybrid Prompting

There is also a possibility to combine several prompting strategies and apply hybrid prompting to achieve better performance of the model (Breve et al., 2024). For instance, one-shot strategy (providing one example) and negative prompting (providing several words or ideas that the model should avoid in the reply) can be used together to guide the model to get a more precise response:

1. Write a short description for advertising new sports sneakers. Do not mention the words “comfort” or “affordable”
2. Example: “Elevate your performance with these high-tech running shoes, designed for speed and agility”
3. Now write one for a new pair of basketball sneakers (OpenAI, 2025c)

2.2.2 Soft prompting strategies

Unlike hard prompting, which directly instructs the model with explicit queries, soft prompts operate by modifying the embeddings and rely more on the LLM's performance, since its query is less explicit. It does not require hard textual prompts, but the model rather learns how embeddings are implicitly suggested by the user. The model is trained on a specific data set modified for each task. Once trained, the learned embeddings are inserted at the beginning of an input sequence, influencing the way the model generates responses (Bhaila et al., 2024).

Prefix-Tuning (Li & Liang, 2021): This method adds trainable prefix vectors to the input while keeping the model's parameters frozen. It’s computationally efficient and effective for domain-specific tasks like legal or medical AI applications. Prefix-tuning is efficient, preserves the base model, and is effective in domain adaptation. Its drawbacks are that it is task-specific, requires training, and has limited generalisability. It is most suitable when adapting large pre-trained models to domain-specific tasks with limited labelled data.

Prompt-Tuning (Lester et al., 2021): This technique optimises continuous prompt embeddings that modify the input prompt. It's useful for fine-tuning models in multilingual contexts or creative applications, without needing task-specific labelled data. Prompt-tuning supports multilingual and creative tasks, enables low-cost fine-tuning, and does not alter model parameters. Its limitations include requiring separate training for each task and limited cross-domain generalisation. It is particularly suitable when fine-tuning is needed for creative, multilingual, or specialised tasks without modifying the base model.

These prompting strategies are efficient since they do not require any modification in the original model, but they are task-specific and demand training, while hard prompting strategies are less efficient since they rely on manually crafted textual prompts that may not generalise well across different tasks.

2.3 Linguistic Impact of Different Prompts

The structure of the prompt can have a profound influence on the response provided by the model. Syntactic complexity, style and wording can have an impact on the possible output, from the point that differently written prompts can be interpreted by the model in other ways.

The research by Linzbach et al. (2024) observed that incorporating supplementary domain and range information into prompts yields a smaller performance improvement when conveyed through appositives rather than clauses. That paper also discovered that adding information via conjunction is less effective than specifying it in isolation (Linzbach et al., 2024).

Moreover, the mood, tense, aspect and modality can highly influence the model's performance. The analysis performed by Leidinger et al. (2023) states that to optimise the performance of LLMs, it is recommended to rephrase all the instructions to questions or orders instead of statements. Authors also specified that even though there was an expectation that usage of passive voice and tenses other than present can potentially lower productivity, it was found that this statement can not be supported, and no major differences were observed (Leidinger et al., 2023). The unexpected result of the research was the fact that using various synonymous expressions can enhance the results even in cases when the synonym used is less frequent: the example provided shows that utilising the words “appraisal” or “commentary” instead of “review” can lead to better output results (Leidinger et al., 2023).

The application of search using LLMs is also influenced profoundly by the possible mistakes made by users. The study by Vadlapati (2023) highlights that spelling and grammar mistakes of different types can result in less accurate responses, even leading to a full misunderstanding of the original question (Vadlapati, 2023).

One more linguistic feature that can potentially enhance the output is the avoidance of using negated prompts, since, according to the paper by Jang et al. (2022), the application of negated prompts results in lower performance compared to the original ones (Jang et al., 2022).

Furthermore, even though a machine can not understand language in the way that humans do, it can also perform differently based on the politeness level of the query. It was discovered that more polite language can not affect the accuracy of the response, but it can influence the length of the output, while some models can even shorten the reply based on the low politeness levels (Yin et al., 2024).

Additionally, the tone embedded in a prompt has been shown to shape the resulting output in systematic ways. The research by Bardol (2025) has shown that when users employ negatively worded prompts, the model tends not to reflect this negativity in the output; vice versa, prompts originally constructed neutrally or positively rarely generate negative replies, which suggests an inherent resistance to downward emotional shifts (Bardol, 2025).

2.3.1 Ambiguity in Prompt Engineering

Semantics plays a fundamental role in prompt engineering since it shapes the quality and relevance of model outputs. The precision of language in prompts is crucial: ambiguities in language can lead to misinterpretations (Chen et al., 2025), making the query unclear for the model, which often leads to responses that do not match completely with the original prompt. The example of the prompt that can look ambiguous to the model can be the one from Tang et al. (2025): “mirror effect in small room”; this query can be interpreted in several ways, and can result in the following Clarification Questions (CQ):

"CQ1. Are you looking for a scientific explanation of how mirrors affect human perception of space? CQ2. Would you like to explore the impact of mirror placements on brightness in your room? CQ3. Do you want to know about the role of mirror shapes in mirror effect? CQ4. Are you interested in tips of small room interior design? CQ5. Would you like interior design tips that could help maximize the space in your room?" (Tang et al., 2025)

Sometimes, instead of asking some CQ model just provides the output that was not requested by the user.

Moreover, approaches such as Semantic Prompt Evolution (SPELL) further highlight the role of semantics in continuously improving prompt effectiveness and model response quality (Li & Wu, 2023). As LLMs continue to evolve, the integration of semantic insights into prompt design will remain critical for refining AI capabilities.

2.4 API prompting

API prompting is a method of interacting with LLMs through an Application Programming Interface (API), allowing for structured and automated input-output processing. It enables several systems to interact with each other. This structured approach allows developers to integrate LLMs into applications while maintaining control over responses.

When using an API for prompting an LLM, the query is sent to a model in a more structured, readable way for a machine, typically JSON, that requires the specification of several parameters, like role (system, user, assistant) and context, which help shape the model's response (Song et al., 2024). Fine-tuning of the output is also possible via determining response length; sampling methods, like temperature (to control response randomness) and top-p (which adjusts probability distribution); and stop conditions to manage response completion. It is important to differentiate this from model fine-tuning, which involves training an LLM on custom datasets to improve performance on specific tasks.

Moreover, using an API for LLM interaction comes with practical considerations, such as rate limits, token usage costs, and latency, which can affect real-world applications.

This approach also has some limitations, including rate limiting (restricting the number of API calls within a specific timeframe), data payload limits (which constrain the size of a payload sent in a single request), latency issues (affecting API response time), and potential costs in large-scale implementations (Serbout et al., 2024; Nilsson & Yngwe, 2022; Manoharan, 2024).

2.5 Evaluating Prompt Effectiveness

Prompt effectiveness can be evaluated based on different parameters, such as relevance, accuracy, consistency, efficiency, readability, coherence and user satisfaction score (Anam, 2025).

Relevance refers to the relation of the query and the answer and how well the response aligns with the user's intent or the context provided. It ensures that the generated output addresses the main question or task without deviating from the topic. Accuracy concerns more with the information in the output: factual correctness and avoidance of hallucinations, which might be especially critical for applications like legal document drafting or healthcare. Consistency means the possibility of reproducing a similar output with the same prompt more than once without significant changes or mistakes in it. Readability & coherence refers to the logic of the output, and can be assessed through readability formulas (like Flesch-Kincaid) or by evaluating sentence structure, grammar, and overall flow manually. The User Satisfaction Score measures users' contentment with the output, typically collected through feedback tools such as surveys or rating systems integrated into the application.

Prompt effectiveness can also be evaluated using metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) or METEOR (Banerjee & Lavie, 2005) or via Automated Evaluation Tools (Ramamoorthy, 2025), such as PromptLayer (PromptLayer, 2025), Azure (Microsoft, 2025a), LangSmith (LangChain, 2025), OpenAI Playground (OpenAI, 2025e) or Langfuse (Langfuse, 2025).

Human Evaluation Metrics can also be applied to assess fluency, relevance or informativeness that can not be evaluated by the machine, though human evaluation is highly subjective, slower and more expensive than machine evaluation and needs standardisation (Chaganty et al., 2018).

2.6 Challenges in Prompt Engineering

The most common challenges one can face using LLMs are considered to be biases, hallucinations, data privacy and lack of output variability (Lin, 2024), as well as language sensitivity. Formulation of prompts in other languages that are not English can lead to a possible lowering of the understanding of the query by the model, since it is trained mainly on the data written in English (Nguyen et al., 2024; Liang, 2023; Kmainasi et al., 2024). This raises ethical concerns, particularly in applications involving hiring, legal decision-making, and medical diagnosis.

Since it is almost impossible to reduce the biases on the training stage, some scholars suggest using strategies to avoid biases using specialised prompting (Kamruzzaman & Kim, 2024) to avoid potentially harmful responses. It can help reduce discriminatory patterns and improve fairness. Additionally, ongoing efforts in AI governance and transparency aim to establish ethical guidelines for prompt engineering, such as “Ethics guidelines for trustworthy AI” (European Commission, 2019), “Recommendation on the ethics of artificial intelligence” (UNESCO, 2021), or “Principles for the ethical use of AI in the UN system” (United Nations, 2022), that ensure AI applications principles alignace with societal values.

One more challenge that has to be considered: task-specific prompt designing (Singh et al., 2024; Chen & Istead, 2024), which is still an effortless strategy since it does not require any additional training. Since LLMs often reflect biases in training data, prompts should be crafted to avoid reinforcing stereotypes or generating inappropriate replies. Specially designed prompts can help guide models toward more neutral, inclusive outputs.

The issue of hallucinations is particularly concerning in domains where accuracy is critical, such as journalism, finance or academia. Prompt engineering strategies such as fact-checking prompts, retrieval-augmented generation (RAG), and source verification mechanisms have been proposed to minimise hallucinations (Shuster et al., 2021).

LLMs can be highly sensitive to word order and slight changes in prompts, which sometimes leads to inconsistent outputs (Zhao et al., 2021).To address this, researchers have explored ensemble prompting techniques, which involve running multiple variations of a prompt and averaging the responses to improve consistency. Additionally, prompt calibration methods help refine input structures to achieve more predictable outputs across different tasks.

Prompt engineering strategies can not only reduce biases but also help mitigate responses that might impact a person’s physical state, particularly in healthcare. After the emergence of LLMs, people have increasingly turned to them for medical advice, seeking symptom analysis and potential treatments. This raises serious concerns about health risks, as AI-generated instructions may not only be incorrect but also potentially harmful. While strategies exist to minimise such risks (Topol, 2023), they cannot be eliminated. It highlights the possibility of applying prompt engineering strategies to multiple fields, including computational linguistics, psychology or design. Its interdisciplinary nature makes it a crucial element in AI interaction and understanding.

2.7 Future Directions of Prompt Engineering

The future of prompting is now shifting towards automatic prompting tools, to make it more personalised (Li et al., 2024b), reducing the interaction of the user in prompting production. For instance, tools like AI Builder can not only help create prompts based on predefined input variables but also test them to provide the most suitable options for the desired query (Microsoft, 2025b).

One more important issue refers to mitigating biased outputs by creating more specific queries (Kamruzzaman & Kim, 2024; Xu et al., 2024). Changing the training data is impossible due to the fact that now models are trained mostly on web data, which is still present with biased and prejudiced representations. Using fine-tuning can be considered this way, but new biased information is still produced, even though nowadays, inclusivity is highly promoted in every sphere.

Recent versions of generative AI began using multimodal prompts that combine multiple input formats, like images, videos, text files or voice messages. It opens various opportunities for users to apply it in image classification, handwriting recognition and translation (Sapkota et al., 2025; Liang et al., 2025; Humphries et al., 2024). This application of LLMs is not yet perfectly developed; therefore, it can be improved.

Moreover, it is worth mentioning collaborative prompt engineering tools, such as Latitude or PromptLayer (Miguelañez, 2025), where multiple users or even AI systems co-create prompts. Such tools enhance the prompt creation process by facilitating teamwork and ensuring more effective and adaptable AI interactions.

One more possible direction of improving prompt engineering experience for users is to analyse users’ behaviour and preferences to create prompts that evolve with the user's interaction style. User-Centric Design can be applied to improve prompt design and performance of the models to ensure a more personalised approach (Mason, 2023).

Chapter 3: Lexical and Syntactic Characteristics of AI-Generated Texts

Despite the impressive capabilities of LLMs in text production, sometimes content generated by AI exhibits features not typical of human-written speech, though resembling it in a genuinely impressive way. AI texts may show particular patterns distinguishing them from those composed by humans, such as peculiar choice in lexical or syntactic structures, discourse incoherence, and pragmatic use dissimilar from human-authored texts. Understanding these characteristics is crucial for applications such as authorship attribution, automated text evaluation, and, moreover, it enhances the quality of machine-generated content.

This chapter provides an analysis of lexical and syntactic features of AI-generated texts, examining their discourse and pragmatic aspects, as well as underlining stylistic and genre-specific variations.

3.1 Lexical Characteristics

AI-generated texts often demonstrate distinctive lexical patterns that set them apart from human-authored writing. Shao et al. (2019) observed that neural methods for data-to-text generation struggle to produce long and diverse texts, often resulting in insufficient modelling of input data and capturing of inter-sentence coherence (Shao et al., 2019). One of these features is frequent usage of high-frequency words; this tendency can be explained based on the mechanism of prediction of the upcoming word of LLMs, which chooses the word that is statistically most probable continuation (Kobayashi et al., 2023). Hence, this feature can be used by automatic AI-detecting tools to check the probability of the text being machine-generated.

In addition, AI-written content can include peculiar or mismatched word choices, especially when contextual depth and specialisation are required. Although such texts may initially appear coherent and even advanced, they can lack semantic precision or relevance (Juzek & Ward, 2025a). The overuse of generalised buzzwords, jargonisms and vague terms further contributes to a lexical style that may sometimes seem too general or vice versa, formal (Juzek & Ward, 2025b; Cabanac et al., 2021). Shao et al. (2019) observed that neural methods for data-to-text generation struggle to produce long and diverse texts, often resulting in insufficient modelling of input data and capturing of inter-sentence coherence (Shao et al., 2019).

3.2 Syntactic Characteristics

From a syntactic perspective, AI-generated texts tend to be highly accurate both in grammar and spelling. It typically follows grammatical rules and rarely contains agreement errors, misplaced punctuation, or typos (Zindela, 2023). While this precision may be viewed as a strength, it can paradoxically serve as a giveaway; human writing often presents features of individual style that can contain small inconsistencies or idiosyncratic phrasing that can only happen with generative AI using specific prompting techniques.

AI-generated sentences can often be seen as monotonous due to repetitions of rhythm and structure; although grammatically flawless, the absence of intentional variation or stylistic risk can render the prose mechanical or overly predictable (Shalevska, 2025; Fedoriv et al., 2023). In contrast, human writers frequently alternate sentence length and construction to create emphasis, build narrative flow, or achieve rhetorical effects.

3.3 Discourse and Pragmatic Considerations

AI-generated texts often fail to exhibit communicative intent or awareness of the audience. Though the models produce syntactically and semantically plausible responses, they do not truly grasp underlying motives and lack sensitivity to nuanced communicative cues such as politeness, tone, or implicatures. This can result in content that is technically accurate but pragmatically inappropriate, particularly in contexts requiring irony, inference, or emotional resonance.

For instance, AI systems frequently struggle with figurative language, including idioms, metaphors, and sarcasm. These forms of expression depend on shared cultural knowledge and non-literal interpretation, which current language models can not fully comprehend. Research by Liu et al. (2022) demonstrated that LLMs underperform significantly when tested on figurative language tasks, such as recognising metaphors or interpreting implied meaning, especially in zero- or few-shot learning settings. Similarly, Jhamtani et al. (2021) found that the presence of figurative expressions in dialogues reduced model performance, and the usage of literal equivalents in place of figurative language improves the performance of the model (Jhamtani, 2021). This inability to interpret non-literal speech reveals a deeper lack of shared human experiences, contextual knowledge, and emotional intuition (Liu et al., 2022). While AI may reproduce figurative phrases through pattern recognition, it does not interpret their underlying meanings or respond appropriately in context. This limitation further restricts

AI’s pragmatic competence, particularly in dialogue and narrative contexts where figures of speech convey the central meaning.

This is particularly evident in translation tasks, where AI tools often fail to preserve the meaning of figurative expressions (Oni, 2025), especially when translating to or from less-represented law-resource languages less represented in the training data. Research by Obeidat et al. (2024) based on the analysis of the idiom translation from English into Arabic has shown that models used face significant challenges in translating figurative language, and need the support and assistance of human translators (Obeidat et al., 2024)

3.4 Stylistic and Genre-Specific Differences

Stylistically, AI-generated texts frequently lack personality or emotional tone. Unless prompted with very specific instructions, they default to a neutral and impersonal register, lacking empathy typical of human writers (Kleinberg et al., 2024). This is especially noticeable in writing that benefits from subjective voice, such as opinion pieces, creative storytelling, or persuasive argumentation. In the research by Kleinberg et al. (2024), it was notices that if the model was specifically instructed to behave more human-like, it still contains some features that can help to distinguish between generated and human-written texts, such as “aplenty with long words, somewhat rare phrasing (“dear friend”) and generally contained less frequent vocabulary (i.e., rarer words)” (Kleinberg et al., 2024).

When attempting to replicate specific genres, AI may successfully imitate surface conventions, such as the structure of an abstract or the tone. In academic writing, for example, an AI might reproduce formal language anf produce better essays than humans do: a study by Herbold et al. (2023) stated that models like ChatGPT outperform humans in generating argumentative essays, and suggested that some actions should be made in the educational sphere to integrate AI technology into the study process (Herbold et al., 2023).

3.5 Automatic AI detection tools

As AI-generated text becomes more prevalent, researchers and developers have created specialised tools to identify whether a piece of writing is AI-generated or human-written. However, detecting AI-generated content remains a challenging task, as the quality of such text continues to improve with advances in machine learning. Recent studies have highlighted the limitations of current detection tools, noting that their accuracy can vary significantly depending on the AI model used and the nature of the text. For instance, some detectors have shown higher sensitivity in identifying content generated by earlier versions of AI models, such as GPT-3.5, but struggle with more advanced models like GPT-4 (Elkhatat et al., 2023). Moreover, the emergence of adversarial techniques, where AI-generated text is intentionally modified to evade detection, further complicates the reliability of these tools (Zhou et al., 2024b).

One prominent example of such tools is OpenAI's AI Text Classifier (Kirchner et al., 2023), which uses machine learning algorithms to distinguish between human and AI-generated text. It is specifically trained on data generated by models like GPT-4 (OpenAI, 2023) and focuses on recognising patterns in phrasing, redundancy, and unnatural language use that often characterise AI-generated content. Despite its effectiveness, it is still far from perfect and has a relatively high rate of false positives when distinguishing between AI and human-written text.

3.6 Key Findings from Previous Studies

Several studies have investigated the strengths and weaknesses of LLM-generated text using corpus-based analysis, revealing both promising capabilities and significant limitations: Schepens et al. (2023) investigated how LLMs could be applied to the creation of linguistically relevant corpora for young German readers, focusing on their ability to reflect word frequency effects, underlying the low richness of the corpus generated by ChatGPT; the study by Uchida (2024) noted that though the ChatGPT-made corpus listed many high-frequency items, it sometimes repeated entries and lacked precision. This approach to AI-driven corpus generation opens new possibilities for the study of language, but also introduces new challenges in terms of evaluating the accuracy and representativeness of such models.

3.6.1 Lexical and Syntactic Patterns

LLMs frequently rely on high-frequency words and simpler syntactic structures, resulting in less lexical variety compared to human-written text, which was proved by the research by Nikolova-Stoupak et al. (2024) conducted in producing sentences based on differences in CEFR levels of language for studying purposes found that LLMs often create structures that are simpler and show less complexity than the ones web-crawled or used as reference in the research. Examples provided by LLMs are also generally shorter (Nikolova-Stoupak et al., 2024). In the same study, the most hapax legomena were found in the web-crawled corpus and the fewest in the LLM ones, showing that the same vocabulary units were used more times in LLMs' generated corpora. While LLMs can generate text with reasonable fluency, they lack the stylistic richness and syntactic variety often found in human writing.

LLMs can be applied successfully to various spheres in an educational context and can support language learning. The corpora mentioned in the previous paragraph can be used to design exercises or comprehension tasks for learners at different proficiency levels. Although promising, such applications raise questions about the pedagogical value of simplified structures and the potential loss of authentic structures and vocabulary found in naturally occurring learner data.

The study by Reinhart et al. (2024) made a significant contribution to the stylistic and lexical analysis of the LLMs' outputs, making specifications that, for instance, GPT models avoid the usage of clausal coordination, while Llama models use it even more frequently than humans; or that in GPT models' output there were discovered higher usage rate of downtoners (such as barely or nearly), while Llama models tend to avoid them. One more vocabulary-related finding that was suggested in this work was the implementation in the outputs of LLMs of the words that typically refer to belletristic works of fiction, in diverse genres (Reinhart et al., 2024).

An interesting finding is that LLMs can adapt the language used to the characteristics of their conversational partner, as it usually happens in human face-to-face interactions (Kandra et al., 2025). Blevins et al. (2025) also found that LLMs tend to mirror the linguistic style of their human partners, at times displaying even greater convergence than humans themselves (Blevins et al., 2025). This suggests that LLMs are capable of producing content that is contextually relevant and can be adjusted for various applications based on the required characteristics, ranging from adapting formality in professional communication to aligning tone and vocabulary in educational, therapeutic, or creative contexts.

3.6.2 Pragmatic and Discourse Handling

The model's accuracy in performance is significantly influenced by the type of linguistic cues provided: for instance, when prompts contain positively framed hints with a "higher lexical overlap" to the target answer, performance improves; on the contrary, introducing contrastive hints leads to a consistent drop in accuracy (Settaluri et al., 2024). These findings point to an underlying reliance on surface-level textual patterns, raising critical questions about the depth of inferential understanding in current LLM architectures. Moreover, Settaluri and colleagues highlight that such sensitivity to prompt phrasing demonstrates how models often succeed not through genuine reasoning, but by exploiting statistical regularities in the input. In other words, the presence or absence of certain linguistic markers can act as shortcuts, guiding the system toward a preferred response without engaging in deeper semantic processing (Settaluri et al., 2024). This raises concerns about the robustness and generalisability of LLMs, as their performance may fluctuate substantially depending on subtle lexical or structural variations in the prompt.

Several researchers have also pointed out that models often struggle to interpret the meaning of the indirect response, while direct ones are more understandable to the model: models sometimes need more context, while human beings can get the meaning from a few words and do not require any additional information (Koo et al., 2025; Settaluri et al., 2024).

At the discourse level, the findings suggest that LLMs struggle to maintain consistent personality traits in long interactions: this highlights limitations in their ability to uphold coherent and stable discourse identities, as personality often fades or shifts depending on context and interaction length (Bhandari et al., 2025).

Uchida (2024), in the study titled “Using early LLMs for corpus linguistics: Examining ChatGPT’s potential and limitations”, observed that LLMs face challenges in genre classification tasks. Specifically, when tested on genre identification, the models achieved low accuracy rates, with only 26.25% accuracy at the word level and 11.25% at the passage level. These results highlight the difficulties that arise when attempting to classify genres using LLMs with precision.

Another challenge that users can face when interacting with LLMs is their poor ability to reason. In the study by Bang et al. (2023), it was mentioned that without explicit prompting, LLM performs poorly in induction (0 out of 30), and achieves slightly better performance in deduction (19 out of 30); but when explicitly asked, induction reasoning increases to 20 out of 30 (Bang et al., 2023). These findings underline the necessity to structure the prompts properly for the LLMs to demonstrate their full capabilities, since without the proper input, it would be difficult to achieve the desired result.

3.6.3 Biases and Errors

It is quite well-known that since the training data is difficult to check, LLMs are prone to producing biased and hallucinated outputs. LLMs have been found to reproduce biases present in their training data, including gender, racial, and cultural biases (Bolukbasi et al., 2016), and also create biases based on the input. Based on this division, biases in LLMs can be classified into two broad groups, intrinsic bias and extrinsic bias (Guo et al., 2024). Intrinsic biases come from the training data; for instance, models often stereotypically associate occupations like “doctor” with men, while “nurse” is associated with women (Bolukbasi et al., 2016; Zhao et al., 2018). Extrinsic biases are based on the application of the model to real-world tasks, which are caused by the data provided by the user or how they interact with the model (Guo et al., 2024). For example, research by Kiritchenko and Mohammad (2018) highlighted that sentiment analysis systems could produce biased results when applied to the analysis of texts that are associated with different demographic groups.

Models can also generate factual errors, particularly when engaging with complex or niche topics. This issue is quite important for users who rely on the models’ judgment without the critical evaluation of the output (Uchida, 2024). However, even knowing that models can not perform ideally, they still can be used for “gaining a general overview or supporting language learning” (Uchida, 2024).

One more issue is that over time, models can become less accurate because the data on which they were trained can become outdated quickly, which is called model drift. There are two types, like “data drift” (when the input data changes) and “concept drift” (when the meaning of the data changes) (Bayram et al., 2022; Mannapur, 2025). As a result, it is crucial to regularly update or retrain models to maintain their performance. Outdated model architectures may no longer be suitable for tasks that require current or dynamically changing information.

A key concern highlighted in the analysis by Curry et al. (2023) is that models tend to take the information that is unrelated to the current question from other concordance lines. This process can lead to the generation of outputs that appear coherent and evidence-based but are, in fact, inaccurate or misleading. One example provided by the authors of the research is an instance that involves the model constructing the reply that implies a discriminatory connection between Islam and homosexuality: even though this information is supported by evidence provided by the model, the supporting material did not refer to Islam and bore no resemblance to known Islamophobic stereotypes (Curry et al., 2023).

When using LLMs in languages that are less represented in the training data, so-called low-resource languages, it is more probable that hallucinations and biases can emerge. In the study by Bang et al. (2023), it was specified that when using LLMs to translate content from high-resource languages like French or Chinese to another high-resource language like English, they generate translations that have better quality than those from low-resource languages like Javanese and Sundanese. In that case, LLMs tend even to generate "words/phrases and sometimes even hallucinate some objects" (Bang et al., 2023), and it was mentioned that sometimes it even translates English text to different languages, even though related to the target ones. These findings suggest that LLMs tend to generalise outputs and raise a question about the applicability of LLMs to generate content in low-resource languages (Bang et al., 2023).

An important consideration in the corpus-based approach of LLMs application and their application in general is the idea of using various prompting strategies to reduce the possible biases, errors and hallucinations, such as zero-shot, few-shot and other prompt engineering techniques. These methods will be discussed in detail in the next chapters.

3.7 Overview of Corpus Linguistics in AI Studies

Corpus linguistics is "a discipline that utilises computer resources to analyse and understand the patterns and variations in language, leading to the development of new theories of language. It enables translators, language learners, and linguists to conduct sophisticated investigations using web-based corpus studies" (Hunston, 2006).

By systematically analysing corpora, researchers can explore patterns in vocabulary, syntax, discourse, pragmatics and other spheres of linguistic analysis. The emergence of LLMs gave linguists the possibility to use AI tools in their experiments, using corpus linguistics to evaluate how these models generate human-like text.

The significance of AI applications to corpora research lies in its ability to "automate and assist in complex tasks that would otherwise require significant manual effort or specialised expertise" (Zappavigna, 2023). The example of such a task can be automated morpho-syntactic annotation, using tools like CLAWS (UCREL, n.d.), or even the application of models to the process of pre-selelection and pre-annotation of the most informative sentences for the selected task (Cañizares-Díaz et al., 2021). Models can also be useful in fact–checking (Hanselowski, 2019) or argument mining, including retrieving argumentative content across entire corpora using indexing plus iterative annotation (Ein-Dor, 2019). Models still require human expertise in determining what is useful and relevant for the specific research, but they can facilitate the process.

One concern that arose after the significant increase in the usage of generative AI is the constant growth of AI-generated content online, which could soon outweigh authentic human-produced language. It may introduce the patterns of language and vocabulary usage that are not typically present in the language, making it harder to compile examples of natural language use, leading to irrelevant research results. Some corpus-based studies also investigate the ability of LLMs to produce fully new text and test if they only apply the memorised patterns from training data and change them. According to Muñoz-Ortiz et al. (2024), human writing shows greater variability in sentence length, also contains richer vocabulary, different dependency and constituent structures, shorter constituents, and more efficient dependency distances; emotional expression also differs, as humans display stronger negative emotions, while AI outputs contain more numbers, symbols, auxiliaries, and pronouns (Muñoz-Ortiz et al., 2024).

Another concern that is frequently raised concerning LLMs is their tendency to forget new knowledge when learning new information. While they may demonstrate perfect memorisation on new tasks, previously acquired knowledge gradually diminishes as additional tasks are introduced (Zheng et al., 2024).

3.8 Implications for Research and Applications

The increasing presence of AI-generated content introduces critical considerations for research, pedagogy, content creation, advertising and other spheres. From a computational linguistics perspective, identifying and interpreting linguistic markers of artificial authorship can deepen our understanding of language modelling and its limitations. It also raises important methodological issues concerning authorship verification, plagiarism screening, and linguistic evaluation. In applied contexts such as journalism, education, and digital media, the ability to distinguish AI-generated writing from human expression is essential for maintaining credibility and ethical standards.

Moreover, AI-authored texts can pose risks related to factual accuracy, contextual sensitivity, and reader trust. As these systems often generate convincing yet erroneous information, critical reading skills are becoming increasingly vital. Future research should explore integrative models that combine automated detection with human oversight to address these emerging challenges in content authenticity and accountability.

The observed lexical and syntactic tendencies in AI-generated texts raise the question of how these outputs align with natural language corpora. To address this, the next chapter introduces methodological approaches that allow for systematic corpus comparison, focusing on both quantitative and qualitative measures.

Chapter 4: Methodological Approaches for Corpus Comparison

This chapter depicts the principal methodological approaches to corpus comparison: quantitative approaches, qualitative approaches, and tools based on NLP. Researchers apply these techniques to test and compare texts based on various characteristics, such as fluency, grammaticality, coherence, factuality, or style. As natural language generation (NLG) systems become increasingly sophisticated, comparing AI-generated content with human-written texts requires a methodological framework that can explore not only linguistic precision but also contextual appropriateness.

4.1 Quantitative Methods

Quantitative methodologies provide a means of comparing corpora based on measurable linguistic properties (Gries, 2025; Lüdeling & Kytö, 2008). These include token and type frequency, dispersion (how elements are distributed), association (how the presence of one element affects the probability of occurrence of another element), etc. (Gries, 2025). By employing statistical tests and algorithmic metrics, researchers can identify and interpret significant differences between human and machine-generated texts.

4.1.1 Fluency and Grammaticality

Fluency refers to the extent to which generated language is smooth, natural, and consistent with native speaker usage (Crystal, 1987, as cited in Chambers, 1997, p. 421). One of the most widely adopted metrics for fluency evaluation is perplexity (PPL), which is the "inverse probability of the test set (one over the probability of the test set), normalized by the number of words (or tokens)" (Jurafsky & Martin, 2024), and it captures how the model is good at assigning probabilities to the text; a lower perplexity score means that the model is able to predict word sequences more effectively, which typically reflects higher fluency of the model output (Jurafsky & Martin, 2024). The formula to compute perplexity is the following, where P is probability, and $W(w_1 w_2 \dots w_N)$ is a test set:

$$perplexity(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}}$$

In addition to automated metrics, human evaluation is also applicable to test the ability of models and to perform a manual check for errors. This approach, though, is more time-consuming and has a higher cost, but it can be useful. It can be applied in different ways, for example, to the sphere of machine translation. The paper by Ramírez-Sánchez et al. (2022) discusses three ways to which human annotation can be useful: (a) manual error annotation of parallel sentence pairs, where human annotators directly identify and mark translation errors in the pairs; (b) post-editing of machine translation output generated by machine translation systems trained on the corpora to indicate the corpus quality; and (c) manual analysis using a concordancer, which involved searching aligned sentence pairs to identify translation inconsistencies.

4.1.2 Semantic Accuracy and Relevance

A central dimension of corpus comparison involves evaluating the semantic coherence of generated texts. Traditional well-known metrics, such as BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation), compare n-gram overlaps between generated and reference texts (Papineni et al., 2002; Lin, 2004). ROUGE evaluates the quality of a computer-generated summary by comparing overlapping units such as n-grams and word sequences with human-written reference summaries (Papineni et al., 2002); BLEU measures how close a machine-generated translation is to professional human translations using n-gram overlap and a numerical score of translation closeness (Lin, 2004).

The more advanced measure that accounts for the paraphrasing and semantic equivalence is BERTScore, which uses "contextualised embeddings from transformer-based models to assess semantic similarity" (Zhang et al., 2020), and not the exact matches like in ROUGE or BLEU. BERTScore can be applied to various tasks, such as text summarisation, translation quality assessment, text generation or document comparison (Zhang et al., 2020).

Corpus comparison also involves assessing the relevance of the generated content based on the relations between the input prompt and output, checking the appropriateness of the generated content and the desired context. Metrics like MoverScore (Zhao et al., 2019) and COMET (Rei et al., 2020) compute token or sentence embedding distances between the source and generated target content, identifying possible overlaps.

4.1.3 Factuality and Hallucination Detection

Since fact-checking and hallucinations have become an important issue in LLMs, factual consistency testing via various automated tools may reduce the limitations of the models so that they can be applied to the spheres where fake information can not be produced due to its high spreading, such as news summarisation, academic writing, or educational content generation. Since models are not able to check their output without a specifically constructed prompt, the credibility of AI outputs needs to be checked differently.

FactCC (Kryściński et al., 2020), QAGS (Wang et al., 2020), and FEQA (Durmus et al., 2020) evaluate the factual consistency of generated summaries of their source texts. They are trained on the annotated datasets and can identify whether the information in the summaries provided lacks supportive claims in the source or if there is even a lack of instances in the original texts. In the same way, benchmark datasets such as FEVER and TruthfulQ serve to measure factual accuracy, based on the truthfulness of the answers generated by the model. FEVER (Thorne et al., 2018) evaluates factual accuracy through evidence-based claim verification, while TruthfulQA (Lin et al., 2022) tests a model's resistance to generating plausible but false answers.

4.1.4 Coherence and Discourse Structure

The coherence of the text, a term which is understood as "relations between small textual units which make the text logically consistent and meaningful to the reader" (Maimon & Tsarfaty, 2023), implies the connection between the words, sentences and paragraphs. It is an aspect that is harder to evaluate, but determining the readability and the effectiveness of the model without it is impossible, since it shows how well the model can replicate human-written content and maintain it throughout the whole text (Zhao et al., 2023).

Quantitative approaches to coherence measurement include models such as the Entity Grid Model introduced by Barzilay & Lapata (2008), which analyses coherence through the representation of a grid, where the rows represent the sentences, and the columns represent the discourse entities. The model records how entities are used syntactically across the text; coherent writing typically demonstrates stable and predictable transitions in entity roles, while incoherent writing features irregular shifts of roles in the text (Barzilay & Lapata, 2008).

Frameworks such as Rhetorical Structure Theory (Wang et al., 2019) or the Penn Discourse Treebank (Prasad et al., 2008) evaluate discourse relations by providing a model of structural connections between the elements of the text. RST models a structure as a hierarchical tree where nodes (parts) are connected by rhetorical relations, labelled as nuclei or satellites to indicate their importance, and discourse coherence is attested by ensuring that every part contributes meaningfully to a complete, supportive structure (Wang et al., 2019). The Penn Discourse TreeBank is a corpus that annotates discourse relations by marking the specific words, called discourse connectives (such as “instead”, “because” or “as the result”) that attach different parts of text (like parts of sentence or paragraphs), and checks coherence by identifying and labelling how different text spans logically relate to each other through these connectives (Prasad et al., 2008).

Measures of lexical coherence, such as word repetition, part-of-speech analysis, and syntactic dependency parsing, can also be applied using various automated tools: libraries like spaCy (Honnibal et al., 2020), Stanza (Qi et al., 2020), and NLTK (Loper & Bird, 2002).

4.1.5 Lexical Diversity and Stylistic Variation

To understand the stylistic richness of the output, it is also important to evaluate the lexical diversity (LD) of the given portion of the text. One of the most common measures of lexical variety is the Type-Token Ratio (TTR), which measures the proportion of unique words (types) by dividing the vocabulary size by the total number of words (tokens) in a given text (Rosillo-Rodes et al., 2025). However, because TTR is sensitive to text length, more complex alternatives such as MTLD (Measure of Textual Lexical Diversity) or HD-D (Hypergeometric Distribution D) are often preferred for longer or variable-length corpora (McCarthy & Jarvis, 2010).

MTLD is calculated as the average length of sequential token strings in a text that maintain a TTR at a preset level (usually .72), obtained by iteratively forming segments from the start and end of the text, resetting whenever a segment’s TTR falls below the threshold, and estimating the final incomplete segment’s contribution (Bestgen, 2024).

HD-D is a lexical diversity index computed using the hypergeometric distribution to calculate the expected TTR for a sample of n tokens by summing the probabilities that each type in the full text appears at least once in the sample and dividing by n (Bestgen, 2024).

Research by McCarthy & Jarvis (2010) confirmed the advantages of these LD measures, showing that MTLD, particularly, is capable of producing stable results regardless of text length and shows a strong correlation with other advanced LD measures (McCarthy & Jarvis, 2010).

Another way to measure the complexity of the text is to apply readability metrics such as the Flesch-Kincaid Grade Level (Ehara, 2024; Imperial & Madabushi, 2023) or the Gunning Fog Index (Yaffe, 2022). The Flesch-Kincaid Grade Level assesses the reading level of the text using the average sentence length and complexity of the words used, and based on these measurements, produces scores that correspond to school grade levels (Tanprasert & Kauchak, 2021). The Gunning Fog Index also takes into account two factors: the length of sentences, because shorter sentences are simpler and easier to read and understand; and the number of complex words with 3+ syllables that are less common in the text with a higher level of complexity (Yaffe, 2022). These scores were primarily developed to be used in an educational context to find the piece of writing suitable for a specific reading level of students. In corpus linguistics, these metrics can also be useful to test if the generated content aligns with the complexity expectations set by the users.

The diversity of the text can also be measured through metrics like Self-BLEU, which takes the average of the BLEU scores for all the generated sentences; lower values of Self-BLEU indicate greater diversity in the generated content (Montahaei et al., 2019). Some researchers suggest using several metrics rather than applying only one to ensure that not only the diversity of the text is assessed but also the quality of the content (Montahaei et al., 2019; Shaib et al., 2025).

4.2 Qualitative Approaches

Though quantitative metrics are able to detect some patterns of the generated texts, some characteristics of the text can be assessed via qualitative methods that find subtle patterns that numerical statistical measures can not observe. Such methods enable researchers to explore how language choices are shaped by context, genre, and communicative intentions (McEnery & Hardie, 2012).

One specific technique frequently employed in qualitative analysis is close and distant reading. Close reading is a strategy that requires more detailed examination of the text, focusing attention on a specific piece of text to check the language of the passage, which would not be possible via automated tools (Mikics, 2007, as cited in Hinchman & Moore, 2013). It is particularly useful in evaluating whether AI-generated texts follow the rules of a specific genre, maintain coherence or do not switch topics. For example, when comparing AI-generated content to that authored by professional writers, researchers might analyse the use of dialogue cues, character voice, and pacing of the story. Distant reading is a technique introduced by Moretti in the book “Graphs, Maps, Trees” in 2005, where he changed the traditional approach of close reading to one that works more with the visualisation of the text into forms of graphs or maps, for instance (Moretti, 2005). Instead of focusing on close examination of words and sentences, distant reading seeks to create a broader perspective by representing overall patterns and characteristics of one or several texts through visualisations (Jänicke et al., 2015).

4.3 Evaluation Platforms

Evaluation platforms are specialised tools designed to assess the performance and behaviour of LLMs (Guo et al., 2023). They provide ways to measure qualities such as accuracy, calibration, fairness or robustness, helping to identify both strengths and limitations of model outputs by integrating automated checks and systematic evaluation methods; these platforms reduce the reliance on manual review and ensure more consistent and transparent assessment across different applications (Chang et al., 2023).

The example of such platforms is, for instance, Azure AI Studio[1]; it offers built-in metrics for assessing groundedness (reliability of sources), relevance (connection of the prompt and the output), fluency (naturalness and readability of the text generated), and safety (checking and mitigating harmful or inappropriate content). These are particularly useful in domains where factual reliability is critical, especially given the ability of generative AI to produce hallucinations and the inability to check the sources used.

The TruLens platform can check truthfulness, question answering relevance, harmful or toxic language, user sentiment, language mismatch, and response verbosity; it uses feedback functions to monitor and assess LLM experiments, reducing the need for manual review and providing a flexible library that developers can adapt to their specific application requirements (Sen, 2023).

4.3.1 Linguistic Annotation and Parsing Tools

Python-based libraries such as spaCy (Honnibal et al., 2020), Stanza (Qi et al., 2020), and NLTK (Loper & Bird, 2002) mentioned are widely used tools for multiple levels of linguistic annotation; these tools support named entity recognition, part-of-speech tagging, dependency parsing, sentence segmentation, text classification, lemmatization, morphological analysis,

[1] https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/observability

entity linking and other NLP tasks. Researchers can use them to extract syntactic structures or entity usage patterns for comparative corpus analysis (Bird et al., 2009).

In addition to Python-based libraries, other libraries do not require a Python environment. For example, CoreNLP (Manning, 2014) is Java-based and also provides the same toolkit that includes tokenisation, sentence splitting, part-of-speech tagging, lemmatisation, named entity recognition, syntactic parsing, coreference resolution, and other annotators (e.g., gender, sentiment) (Manning, 2014). One more example that can be mentioned is UDPipe (Straka & Straková, 2017), which can be used via C++, Python, Perl, Java and C#, and it offers tokenisation, part-of-speech tagging, lemmatisation and parsing (Straka & Straková, 2017). These tools allow researchers without sufficient Python knowledge performing NLP tasks without sufficient Python knowledge.

4.3.2 Experiment Tracking and Data Visualisation

Platforms like Weights & Biases allow the usage of a combination of metrics, such as BLEU, ROUGE or perplexity, to track the outputs of the models, detecting patterns in incorrect or biased outputs[2]. This approach provides a more comprehensive approach to model behaviour by monitoring multiple evaluation metrics, so that researchers are able to observe recurring tendencies in the data, particularly those that reveal underlying biases or systematic issues. Since this platform allows fine-tuning, it enables gradual improvements, supporting the development of models that produce more reliable and consistent contexts.

4.4 Synthesising Methodologies

No single evaluation method is sufficient for a comprehensive comparison of human and machine-generated texts. Instead, researchers increasingly employ multi-method approaches that combine quantitative and qualitative methods, using both automated tools and platforms as well as human annotation. Choosing appropriate methods depends on the type of text and evaluation goal; for instance, in legal and scientific texts, it is more important to maintain a high level of factual accuracy, checking the biases and hallucinations, while creative tasks require more language diversity and coherence. Researchers must also consider ethical implications, particularly in evaluating for bias, toxicity, or misinformation.

[2] https://docs.wandb.ai/guides/

Chapter 5: Research Gap and Justification for This Study

5.1 Identified Gaps in Previous Research

Linguistic corpora have become foundational resources for empirical research in various fields of application, such as psycholinguistics, language acquisition, education and computational linguistics. Among the most widely used ones are the corpora based on the web-resources (called web-crawled corpora), that capture various speech patterns and the rapid change of language; moreover, some corpora are constructed of film scripts or subtitles, like Open Subtitles corpus (Lison & Tiedemann, 2024), or the ones consist of book text for various age groups such as Children Stories Text Corpus (Eden, 2021).

Each of these collections of texts offers specific advantages for linguistic research, but they also have some limitations, such as the limited representativeness of specific linguistic functions.

For example, OpenSubtitles, though rich in vocabulary, structures and dialogue, contains content that is artificially created reflecting informal or cinematic speech patterns, which can include idiomatic, sarcastic, or culturally bound expressions that may not generalise well to broader contexts, and that may not be suitable for pedagogical or developmental purposes. In contrast, the Children Stories Text Corpus focuses on written language targeted for children, but does not capture the dynamic features of spontaneous speech.

Web corpora, such as those presented at the Sketch Engine website (Kilgarriff et al., 2004; Kilgarriff et al., 2014), provide multiple and diverse characteristics, but tend to be skewed towards adult-oriented, formal registers. These corpora often lack detailed data regarding age, audience, or communicative intent, making it also difficult to isolate specific linguistic patterns. Moreover, web corpora frequently include noise, such as non-standard spellings or typos, or irrelevant content, which complicates clean linguistic analysis.

The introduction of generative AI tools into corpus linguistics made it possible to foster a more comprehensive perspective on this field, applying the capabilities of LLMs to generate vast amounts of coherent, plausible text across a wide range of domains. However, the implementation of LLMs in simulating corpus-like language data, especially through the usage of different prompting strategies, has not been studied systematically yet. Research in this field usually concerns the evaluation of coherence, reliability of the sources used, task completion, or the testing for possibility to detect the generated content (van Noord et al., 2024; Cornelius et al., 2024; Ali et al., 2025; Liu et al., 2023; Jiang et al., 2024; De la Iglesia et al., 2025), but not the resemblance of the natural corpora with the ones produced by the model.

One more issue that needs to be addressed is the application of prompt engineering, since few studies have examined how the complexity of prompts affects the linguistic features of the generated corpora (Imperial & Madabushi, 2022; Leidinger et al., 2023; Mu et al., 2024; Murr et al., 2023; Rawte et al., 2023; Sclar et al., 2023; Tang et al., 2024; Wahle et al., 2024). Prompt engineering has been applied in fields such as programming or summarisation, but it remains less explored in the linguistic domain, considering the lexical and syntactic characteristics of the output.

Finally, there is a gap in how the generated corpora are evaluated linguistically, since metrics like BLEU or ROUGE can not capture all the features relevant for the full assessment of the texts, that usually requires more in deep analysis of syntactic complexity, lexical diversity or patterns, which is crucial in terms of evaluation how generated language aligns with human-written content and how it can be applied in various fields like education, law or medicine.

5.2 Relevance of This Study

This study not only discusses and explores the existing gap in corpus-based linguistic analysis but also adds to the recent works that investigate the potential of LLMs in generating synthetic language corpora with comparison to natural corpora (Nikolova-Stoupak et al., 2024; Muñoz-Ortiz et al., 2024; Zwerdling et al., 2022; Schepens et al., 2023; Tudino and Qin, 2024; Berber Sardinha, 2024). In particular, the research by Schepens et al. (2023) application of the LLMs for generating corpora demonstrates how these models can capture phenomena like the word frequency effect for young readers, offering an empirical case for using LLMs in cognitive and developmental linguistics; while this study targets on young German readers with specified age restrictions, the broader methodological approach aligns with the current research, as it also assess the linguistic similarity of generated language and natural language.

Although the corpora in this study are not age-specific, their comparative analysis with corpora such as the Open Subtitles corpus, the Children Stories Text Corpus, and a web-crawled one helps contextualise how generative AI can be evaluated across different levels of linguistic structure and usage patterns.

This comparative approach is significant for several reasons:

Controlled Prompt Variation

By designing three different types of prompts with different levels of complexity, this study provides insight into how prompt structure influences linguistic features in the output. This variation also makes this study an investigation of how prompt engineering can be applied to control the outputs and make them more systematic and replicable.

Benchmarking Against Varied Corpora

While the prompts for generated corpora do not contain any information about the age restrictions, comparing them to the corpora of a certain communicative intent (adult audience for the subtitles corpus, children audience for the books corpus, and varied audience for web-crawled corpus) makes it possible to assess how can the output generated by LLMs approximate different language registers and test whether various complexity of prompts can influence the production of the output that aligns with specific stylistic and structural norms.

Linguistically-Oriented Evaluation

The study employs a set of linguistically grounded metrics to assess the syntactic complexity of the generated corpora versus three chosen corpora compiled by linguists. These include measures such as IDT (Incomplete Dependency Theory Metric) (Zou et al., 2022; Mirzapour et al., 2018) or MLS (Mean Length of Sentence) (Kyle & Crossley, 2018; Liu & Afzaal, 2021; Liu et al., 2023). These metrics are selected to capture meaningful variation in the structure of the texts, independent of task or topic, assessing the complexity of the generated content.

New Corpus Construction Techniques

This work contributes to the quite recent tendency to synthetic corpus construction using LLMs, which can be adjusted for a variety of linguistic analyses that can be adapted for tasks such as psycholinguistic modelling or educational material design.

Theoretical Extension of Prompt Engineering

It was already discussed that prompting strategies can imply regulations to avoid harmful output and hallucinations, and this study contributes to the discussion of prompt framing that can serve as a tool for linguistic calibration of corpus creation.

5.3 Expected Contributions

With the growing tendency of applying computer-generated language across various spheres, such as educational, technological, or social, it also becomes important to examine how the structure and phrasing of prompts influence the quality and characteristics of the resulting text. This thesis offers both theoretical insights and practical findings that aim to support ongoing research at the intersection of automated text generation, corpus analysis, and prompt design.

5.3.1 Theoretical Contributions

This study reveals how LLMs respond to the remodelling of prompting complexity and the impact of the prompt complexity on the form and structure of the output, showing how model instruction influences discourse production.

Even though the generated corpora are not age-specific, their comparison with age-oriented and general-purpose corpora provides insights into how LLMs unconsciously replicate or diverge from established communicative norms.

5.3.2 Methodological Contributions

The study uses a method of generating corpora via Python and the OpenAI API (OpenAI, 2025d), suitable for controlled linguistic experimentation and can be reproduced. It enables researchers to systematically manipulate input parameters, ensuring consistency across experimental conditions while maintaining flexibility for diverse linguistic research goals.

5.3.3 Practical Contributions

The generated corpora can also serve as data sources for computational and educational research. They can be adapted or extended for more targeted uses (e.g., simplified reading materials, test datasets). The results of this study will offer practical guidance on how to craft prompts that lead to desired stylistic or structural outcomes.

Chapter 6: Methodology and Data Collection

6.1 Overview of the study design

The main purpose of this study is to investigate the linguistic characteristics of the small corpus of texts generated by AI and to compare it to authentic human-compiled corpora to check whether generative AI has enough capabilities to produce content similar to human-written one and if this content can facilitate the process of corpus creation so that the generated corpora can be used in various linguistic research. The aim of the study also focuses on the evaluation of how prompt complexity influences the linguistic output of the LLM in terms of structural and stylistic features.

To accomplish this, six corpora were analysed. Three of them were generated using the OpenAI API (o1 model) (OpenAI, 2024; OpenAI, 2025d), with three levels of prompt complexity employed; they are referred to as Group 1 Prompts, Group 2 Prompts, and Group 3 Prompts. The other three are the corpora used as reference, and they are the ones composed of human-created texts: monolingual data from the Open Subtitles corpus (Lison & Tiedemann, 2016), Children Stories Text Corpus (Cleaned Gutenberg books) (Eden, 2021), and two subsets of the Leipzig Web Corpus (Goldhahn et al., 2012).

6.2 Description of Corpora

6.2.1 AI-generated Corpora

Three corpora were created using the o1 version of the OpenAI API (OpenAI, 2025d); the prices for this version are: input tokens: $15.00 per 1 million tokens, cached input tokens: $7.50 per 1 million tokens, output tokens: $60.00 per 1 million tokens[3]. The price for the output generated in this study was approximately $13.28 (word count 165,939; estimated token count 221,252 tokens). The o1 model series was chosen for the study because of its design; the model is trained using reinforcement learning techniques to support advanced chain-of-thought reasoning that enables the model to create more deliberate and context-aware responses[4]. These characteristics make the o1 series particularly suitable for producing linguistically rich and reliable data for comparative corpus analysis.

[3] https://platform.openai.com/docs/pricing
[4] https://openai.com/index/openai-o1-system-card/

To create the corpora, three levels of prompts were made, with three different levels of complexity: from simpler ones to more complex ones. Each of them requested the model to produce a short text that should resemble a film script. The decision to structure the prompts into three complexity levels was specifically made to check how the output of the model changes based on how comprehensive the prompt is. Since the employment of the OpenAI API (OpenAI, 2025d) does not allow the modification of the prompt after the output is produced, it was needed to carefully design all the levels of prompts in advance to ensure they request all the types of information to produce the required form of the output.

The three groups are divided as follows:

Group 1 Prompts: Simple Prompts

These prompts were the shortest and the simplest ones, requesting a film script for a certain topic without any specifications, narrative guidance or stylistic framing; only one sentence in length. They also did not specify the exact number of words that were needed to produce, leaving the choice for the machine: “Invent a film script that focuses on journalism, investigative reporting, or news media”.

Group 2 Prompts: 2000-Word Intermediate Prompts

These prompts also requested a film script for a certain topic, but they gave some hints to the model to ensure better performance, specifying in more detail what subplots the film scripts should contain; they all consisted of two sentences. The main difference between the first and the second group was that the intermediate one required the model to make the output that should be at least 2000 words in length: “Invent a film script that focuses on reportage or field journalism. The selected script should involve journalists actively reporting from real-world events, covering social or political issues. Output should be at least 2000 words.”.

Group 3 Prompts: 2000-Word Advanced CoT/ToT Prompts

The purpose of these prompts was the same as in the first two groups: they requested a film script for a certain topic, but in a more advanced way. They not only specified the length as at least 2000 words, as in the second group, but they were also developed using Chain-of-Thought or Three-of-Thought prompting techniques. Prompts consist of multiple sentences, usually five: the first three sentences describe in detail the topic of the script and details that it should contain, like character description, the development of the story or the contextual descriptions. The fourth sentence included two examples of existing films that can be used as guidance for plot development. The last one only mentioned the required length of the output. They aimed to elicit more nuanced and cohesive outputs by guiding the model through a multi-step reasoning process: “Invent a film script that focuses on journalistic investigations driven by real-world political or social scandals, with a strong focus on uncovering hidden truths. Emphasise stories where the process of gathering evidence and the ethical dilemmas of reporting play a central narrative role. The scripts should highlight the tension between media freedom and institutional power. Examples include: Spotlight, All the President’s Men. Output should be at least 2000 words”.

Each group consisted of 15 prompts that were derived using the LOB corpus sampling frame described by Hofland and Johansson (1982), and referenced in McEnery & Hardie (2012). This sampling strategy ensured a balanced representation of topics throughout the whole corpus to make it representative of linguistic forms and structures across the different prompt types.

Although each of the 15 prompts in each group was tailored to a specific topic, it can be noticed that the same structural template is shared by all the prompts. It ensures that the inaccurate vocabulary choice of the prompts can not have much influence on the production of the output, still allowing sufficient variation to reflect a diverse range of themes and scenarios, guaranteeing that any linguistic differences observed across corpora can be systematically linked to prompt complexity rather than extraneous factors. The 15 topics are: Press: reportage, Press: editorial, Press: reviews, Religion, Skills, Trades and Hobbies, Popular Lore, Belles Lettres, Biography, Essays, Miscellaneous (Government Documents, Foundation Reports, Industry Reports, College, Catalogue, Industry House Organ), Learned and Scientific Writings, General Fiction, Mystery and Detective Fiction, Science Fiction, Adventure and Western Fiction, Romance and Love Story, Humour (Hofland & Johansson, 1982, as cited in McEnery & Hardie, 2012).

The overall prompt design is also deliberately created with balanced levels of instruction: the Group 1 prompts allows for the creativity of the model specifying only the topic without any other restrictions; Group 2 prompts were more structured and hence more shaped outcomes; Group 3 being the longest and the most detailed prompts provided multistep guidance to elicit more coherent and consistent responses; the embedding of real film examples into the prompts was made not just to offer thematic inspiration but also to control the model output providing it with known cinematic styles, narrative arcs, and genre expectations. This contrast made it possible to examine how these implementations can influence the structure, quality and creativity of the output based on different levels of restrictions provided.

The addition of a minimum word count requirement in Group 2 and Group 3 prompts structure served as a mechanism to encourage the model to produce scripts that can be more developed, preventing the production of overly short outputs that can be less representative of the patterns usually presented in written film script, making it harder to make the comparison with the existing human-compiled corpora.

It was predicted that more detailed and well-structured Group 2 and especially Group 3 prompts would encourage the generation of better organised, contextually and thematically appropriate outputs. In contrast, the outputs based on the simplest Group 1 prompts were expected to be more predictable and less accurate.

All the prompts for these groups can be found in the Appendix section.

6.2.2 Reference corpora selection

To compare and contrast the three corpora created by ChatGPT, the three reference corpora were selected to provide a diverse representation of human-written texts and how corpora compiled by professionals are structured. These corpora are: OpenSubtitles English Monolingual Data (Lison & Tiedemann, 2016), Children Stories Text Corpus (that presents cleaned Gutenberg books) (Eden, 2021) and web-crawled Leipzig Web Corpus (Goldhahn et al., 2012):

OpenSubtitles (English Monolingual Data)

Since the corpora that were created in this study by AI present short film script outputs, it was necessary to select one reference corpus that is representative of the same text structures and stylistic characteristics as the generated ones. The OpenSubtitles corpus is originally designed as a parallel corpus that provides the subtitles for films in pairs, but it also gives an opportunity to extract and download only monolingual data from it, which was used in this research to make the comparison with the generated corpora. For this research, the 2024 release of the English-language monolingual subset was used; it contains a wide range of films and television shows. Its suitability as a reference corpus can be explained by its diversity: its content is representative of various speech registers ranging from highly informal conversations to more formal and structured ones, both should be observed in AI-generated scripts since they attempt to recreate human dialogues as well as genre and register-specific conventions. The presence of elliptical structures, discourse markers, interruptions and other features typical of film dialogues ensures that OpenSubtitles captures the specificity of this text genre. One more important feature is that the generated corpora are representative of different themes of films that should also be present in the reference corpus to make it possible to compare syntactic and semantic variations across different domains. Additionally, the corpus is large-scale, pre-processed, and publicly accessible, which enhances its reproducibility and reliability for linguistic comparison.

Since the size of the corpus is too big for parsing and application of metrics, it was decided to sample just a part of the corpus that would be thematically representative and manageable in terms of computational resources, while still preserving the diversity of genres, dialogue styles, and syntactic constructions necessary for meaningful comparison with the AI-generated corpora.

Leipzig Web Corpus

The Leipzig Corpora Collection is an open-access linguistic resource that provides large-scale, preprocessed text corpora compiled from web sources (Goldhahn et al., 2012). Designed for comparative linguistic research, it includes multilingual datasets that cover a wide range of domains and genres. In this study, two English corpora from the Leipzig Web Corpus were used: Web-public UK 2018 and Web-public com 2018. Their inclusion allows for meaningful comparisons with AI-generated texts, particularly in terms of syntactic constructions and lexical diversity. The broad topical coverage and considerable size of these corpora make them suitable benchmarks for assessing the naturalness and variation in the outputs generated by language models. However, as with many web-derived resources, the corpora may contain noise, such as spelling inconsistencies or domain imbalance, which must be taken into account during analysis.

Children Stories Text Corpus (Cleaned Gutenberg books)

The more traditional corpus was chosen to provide a more complex sentence and text structures, and present a broader range of vocabulary that is less colloquial and is not typical of film scripts and web language. Children’s literature often includes rich narrative styles, descriptive passages and unusual usage of vocabulary; authors of such texts can provide neologisms (invented vocabulary) to entertain children and capture their attention, and they tend to employ some peculiar stylistic devices that are not typical in everyday communication, such as rhymes, personifications or repetitions. Furthermore, children's literature is usually created with some pedagogical and moral ideas shaping it, so they tend to feature more structured plots and coherent logical discourse patterns that contrast with the more fragmented nature of film scripts or web discourse. The cleaned Gutenberg selection also ensures that the data in the corpus is free from errors and unrelated data, making it possible to apply it for comparative linguistic analysis.

Together these three reference corpora offer perspective for linguistic analysis form multiple perspectives: OpenSubtitle is the most similar corpus to the generated ones and offers a representation of style and structures of scripted dialogues; Leipzig Web Corpus provides more informal web-based natural language use; and the Children Stories Text Corpus introduces more formal but more literary and narratively rich language with more structured discourse with larger variety of language and stylistic patterns. By selecting corpora that are representative of distinct linguistic domains: dialogues, informal web discourse, and narrative literary ones, the study aims to capture various characteristics of the AI-generated text for multidimensional comparison. Bringing together insights from different genres, this study offers a deeper understanding of how well AI can reflect the richness and variety found in human communication.

Corpus	Source & Compilation	Language & Release	Size & Sampling	Text Type / Genre
OpenSubtitles (English Monolingual Data, 2024 release)	Subtitles extracted from a multilingual subtitle database; originally a parallel corpus, here used as English-only monolingual data	English (2024 monolingual subset)	Very large-scale corpus (hundreds of millions of tokens); only a thematically representative sample used for computational feasibility (32.0 GB)	Film and TV subtitles (dialogue scripts)
Leipzig Web Corpus (Web-public UK 2018 & Web-public com 2018)	Collected from diverse web sources, preprocessed and standardised by Leipzig Corpora Collection	English (two web-based corpora: UK domain and .com domain, both 2018 releases)	Large-scale corpora (tens of millions of sentences); complete datasets used	Web-based texts across multiple domains and genres
Children Stories Text Corpus (Cleaned Gutenberg Books)	Digitised children's literature sourced from the Gutenberg project; cleaned to remove errors and unrelated material	English (selection of children's books), last update 4 years ago (2021)	Medium-scale corpus; manageable size (19.6 MB) due to pre-cleaning; exact token count smaller than OpenSubtitles and Leipzig	Children's literature: narrative prose, storytelling, descriptive texts

Table 2. Characteristics of the Corpora Used

6.3. API Implementation

As it was already mentioned earlier, the corpora generation process was automated by implementing the OpenAI API system that was accessed via downloading the openai library package (OpenAI, 2025d). Each piece of code for each group used the client.responses.create a function in which the model type (o1), the instructions to the model and the input were specified. Each call to the API included a standard instruction string "You are a helpful assistant", to minimise the possible variations in model behaviour. The outputs were stored in plain text files with indexing to each group and prompt, facilitating both human inspection and automated processing.

To organise the generation process the outputs were divided into three separate codes and stored in three separate files, each corresponding to one of the prompt groups; keeping the groups distinct makes it easier to control the possible errors in the output and change the prompts in case of mistakes in the generated outputs; moreover, it is easier to compare the outputs later when they remain in the separate files. Working with separate files also offers time reduction advantages, since generating all three groups in one output can cause significant time delays; breaking the workload into three smaller units helped to control the overall waiting time. This setup also signifies that the outputs were saved in a uniform, well-structured format: each output was linked clearly to its input prompt, and the overall consistency of the dataset made the later analysis easier and more straightforward. Importantly, a fully automated process of output production minimises the need for manual intervention that can cause mistakes and also makes it easier to repeat and verify the whole procedure if necessary.

6.4. Preprocessing of Data

To prepare the output for linguistic analysis, a thorough preprocessing phase was carried out. This step is essential to ensure that the texts are cleaned, structurally consistent and compatible with the implementation of the metrics used in the later stages.

The tool chosen to perform the preprocessing of the text was spaCy (Honnibal et al., 2020), a Python NLP library developed by Explosion AI. This particular toolkit was selected for its high accuracy across required tasks (tokenisation, lemmatisation, part-of-speech tagging, syntactic dependency parsing, etc.) and its compatibility with various languages, including English.

Since the aim of this study is to provide a transparent analysis of AI-generated text, any interventions such as editing or removing fragments of the output were considered potentially damaging to the integrity of the dataset, and altering the original output could introduce biases or artificial patterns. Removing the whole text was also not possible, since the chosen 15 topics represent various vocabulary and syntactic structures based on their thematic diversity and were essential for maintaining balance across the generated corpus.

Chapter 7: Analytical Framework

This study utilises a detailed analytical framework and tools to examine syntactic and stylistic characteristics of AI-generated texts and compare them to the existing human-authored corpora. The framework combines traditional syntactic complexity measures, syntactic processing complexity measures and emotional features of text using a combination of traditional corpus linguistic methods and transformer-based natural language processing.

These dimensions offer a multidimensional and advanced exploration of not only the surface-level grammatical properties but also the processing complexity of text, as well as its stylistic features. Together, they can facilitate the process of comparison of the corpora generated by ChatGPT with those composed by humans.

7.1 Analytical framework

The analytical framework is based on the three components:

- Syntactic Complexity Metrics

Syntactic Complexity Metrics can be applied to measure the structural complexity of the sentences and texts by characterising how the grammatical elements are organised and related to each other. These metrics provide some information on how varied the presented syntactic structures are, checking the sentence length, the sentence types and usage of subordinate clauses.

- Syntactic Processing Complexity Metrics

These metrics are based on the Dependency Locality Theory (DLT) by Gibson (2000), and by applying them, it is possible to evaluate the cognitive capacities required to process and understand the syntactic structure of sentences. According to the DLT, the longer the distance between the dependent elements in the sentence is, the more effort should be spent to process these structures. These metrics assess the complexity of a sentence using such characteristics as sentence length, structure of syntactic trees, and the relation of the syntactic elements nested within the sentences.

- Sentiment analysis (Emotion Analysis)

To test the expressive characteristics of the corpus, it was decided to apply the techniques of sentiment analysis and automated tools that are able to classify the emotional tone in the textual data and test how it is distributed in the corpora generated by ChatGPT.

Each of these dimensions is operationalised through specific metrics and models described in the following subsections.

7.1.1 Syntactic Complexity Metrics

Syntactic complexity can be assessed on different levels, such as clause, sentence, or phrase levels, making it possible to capture the structural richness and language variations across different types of texts and multiple topics. In this study, Syntactic Complexity Metrics are organised into five functional groups according to Lu (2011). Some other studies also adapted this classification to enable systematic comparison and ensure consistency to measure syntactic variations across corpora, such as Kyle & Crossley (2018), Kyle et al. (2021), Liu & Afzaal (2021), and Liu et al. (2023).

The metrics were automatically implemented using the dependency parsing tool provided by Spacy and computed using Python functions.

Some of the terms used in the metrics should be specified; Table 3, adapted from Kyle & Crossley (2018), provides them with some definitions:

Structure	Description
Word	A sequence of letters that is bounded by white space
Verb phrase	A finite or nonfinite verb phrase that is dominated by a clause marker
Complex Nominal	a) Nouns with modifiers b) Nominal clauses c) Gerunds and infinitives that function as subjects
Coordinate phrase	Adjective, adverb, noun and verb phrases connected by a coordinating conjunction
Clause	A syntactic structure with a subject and a finite verb
Dependent clause	A finite clause that is a nominal, adverbial, or adjective clause

Table 3. A Description of Syntactic Structures used in metrics

Each five groups of metrics are described in detail below:

Type 1: Length of Production

- Mean Length of Clause (MLC): It can be counted by dividing the number of words by the number of clauses.
- Mean Length of Sentence (MLS): It can be counted by dividing the number of words by the number of sentences.

Type 2: Sentence Complexity

- Clauses per Sentence (C/S): It can be counted by dividing the number of clauses by the number of sentences.

Type 3: Subordination

- Dependent Clauses per Clause (DC/C): It can be counted by dividing the number of dependent clauses by the number of clauses.

Type 4: Coordination

- Coordinate Phrases per Clause (CP/C): It can be counted by dividing the number of coordinate phrases by the number of clauses.

Type 5: Complex Nominals

- Complex Nominals per Clause (CN/C): It can be counted by dividing the number of complex nominals by the number of clauses.

These metrics allow for the quantification of grammatical richness and provide a basis for comparing stylistic variation across corpora.

7.1.2 Syntactic Processing Complexity Metrics

Unlike traditional metrics of Syntactic Complexity, Syntactic Processing Complexity Metrics focus not on the form of the utterance, but on how the structure of the sentence can influence the process of comprehension of the text. These metrics are used in various research, such as Zou et al. (2022), Rathi (2021), or Mirzapour et al. (2018).

The definitions for the metrics below are adapted from Zou et al. (2022):

- **Incomplete Dependency Theory Metric (IDT):** For a given token t_i, the IDT metric counts the number of incomplete dependencies between t_i and t_i+1.
- **Dependency Locality Theory Metric (DLT):** A token t_i is said to be a discourse referent whenever its part-of-speech tag is a Proper Noun, Noun or Verb.
- **Combined IDT+DLT Metric (IDT+DLT):** For a given token t_i in a sentence, the combined IDT+DLT metric is the sum of the IDT metric and the DLT metric for token t_i.
- **Nested Nouns Distance Metric (NND):** The distance between two tokens t_i and t_j is their absolute positional distance, | j-i |.
- **Left-Embeddedness Metric (LE)** (Cheung & Kemper, 1992), Yngve (1960)**:** The LE metric counts the number of tokens on the left-hand side of the main verb which are not verbs.

In this study, for each of the five syntactic processing complexity metrics, the sum (SUM), average (AVG), and maximum (MAX) values were computed per text. These aggregate values were also adopted from Zou et al. (2022), who employed them to quantify processing load across longer text spans rather than on a sentence-by-sentence basis. This approach allows for a scalable and comparative analysis of syntactic processing difficulty between AI-generated and human-authored corpora.

It should also be noted that the analysis is largely quantitative and static, focusing on metric values aggregated over corpora. This approach does not take into account dynamic aspects of discourse or pragmatic considerations that may shape syntactic choices, nor does it assess semantic adequacy or textual coherence beyond the level of surface structure.

Syntactic complexity metrics, while informative for some analyses, do not account for the semantic richness or pragmatic appropriateness of generated texts. A text produced by an AI system may, for instance, appear semantically sophisticated or contextually relevant (an aspect that was not explored in this study).

7.1.3 Sentiment Analysis (Emotions Analysis)

In addition to the Syntactic Complexity and Syntactic Processing Complexity Metrics, it was decided to also implement the automated classification of emotions to assess not only the structure of the output of ChatGPT, but also how it manages to cope with the complexity of stylistic and expressive variation of human language to produce the content that corresponds to the human speech expressiveness.

To make the sentiment analysis, the Emotion English DistilRoBERTa-base was used (Hartmann, 2022). It can be accessed via Hugging Face. This model is built upon the transformer architecture, which relies on self-attention mechanisms to capture relationships between words in a sentence, regardless of their distance from each other. Transformers employ layers of encoders and decoders that allow the model to process input sequences in parallel rather than sequentially, improving both efficiency and contextual understanding. In particular, the self-attention mechanism assigns weights to different words depending on their relevance to the current word, enabling nuanced interpretation of complex linguistic structures (Vaswani et al., 2017). The model was trained on six diverse datasets and predicts Ekman's six basic emotions, with an additional neutral class, resulting in seven categories: Anger, Disgust, Fear, Joy, Sadness, Surprise, and Neutral (Hartmann, 2022).

Since the model can categorise all the emotions that can not be classified into the “Neutral” group, all the calculations and analyses were made based on the six other emotional categories to ensure that the results accurately reflect the distribution and intensity of clearly defined emotional states without including any ambiguous inputs.

This model enables a deeper exploration of:

- Emotional diversity across texts,
- Dominant emotional tone in different corpora,
- Affective tendencies in AI-generated language compared to human writing.

Chapter 8: Metric Application Results

This chapter presents the quantitative results of the syntactic metrics applied to six already mentioned corpora: three AI-generated datasets (Groups 1, 2, and 3) and three human-written datasets (OpenSubtitles, Children Stories, and Leipzig Web corpora from the UK and COM domains). For clarity of presentation of the results, they are grouped into two main categories: syntactic complexity (clause-based) metrics and syntactic processing complexity (dependency-based) metrics. Each set of results is followed by a description of the numbers and their significance for the research. The detailed interpretation and analysis of the figures will be presented in the next chapter.

8.1 Clause-Based Syntactic Complexity

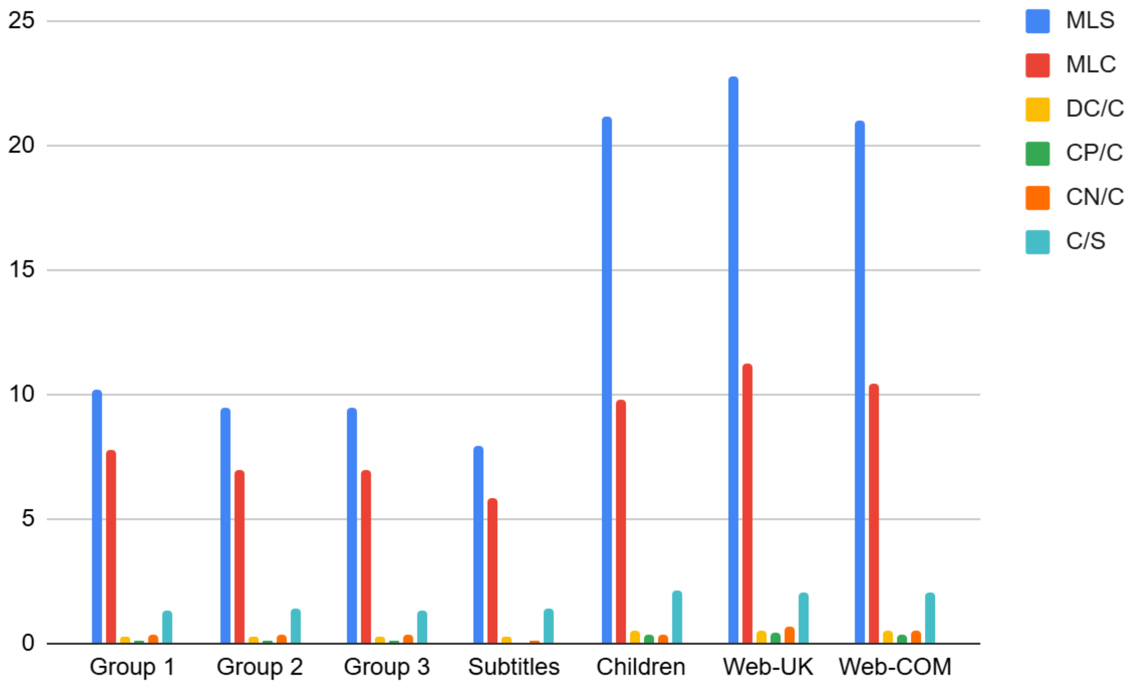

Figure 1. Clause-Based Syntactic Complexity Metrics

The results of the clause-based metrics reveal a consistent distinction between AI-generated corpora and the Subtitles corpus with human-authored corpora. AI groups (1–3) show moderate syntactic complexity, with sentence lengths around 9–10 words and clause lengths between 6.9 and 7.8. Among them, Group 1 has the highest values for both sentence and clause length, suggesting a slightly more elaborate syntactic style. Subtitles feature the shortest sentence and clause lengths, reflecting their informal, conversational nature.

The Children corpus and both of the Web corpora (UK and COM) display significantly higher clause-based complexity levels; sentence lengths exceed 21 words, and clause lengths range from 9.8 to over 11 words. These corpora also demonstrate greater subordination (DC/C), coordination (CP/C), and structural density (C/S). Web-UK stands out with the highest CP/C (0.449) and CN/C (0.648), indicating dense noun phrases and frequent coordination. This suggests that Web texts are not only longer but structurally richer and more embedded.

Group 3 displays a slightly higher CN/C value than other AI groups (0.383), suggesting improved nominal complexity, but it remains well below the values in the Web corpora. Overall, human-written corpora display significantly greater syntactic depth and variation.

8.2 Dependency-Based Syntactic Processing Complexity

To offer a comprehensive view of the data, dependency-based metrics are presented in three parts: AVG values, MAX values, and SUM values.

8.2.1 Dependency-Based Metrics: Average Values

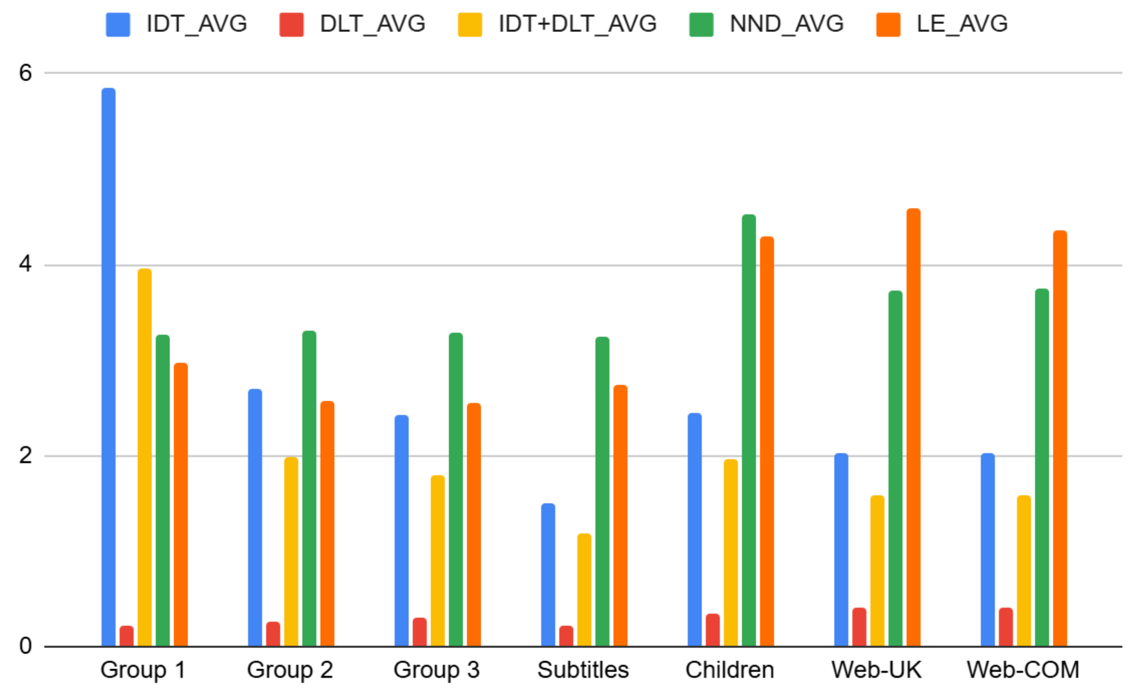

Figure 2. Dependency-Based Metrics: Average Values

The average values reveal that Group 1 has the highest IDT_AVG (5.861), which could reflect repetitive nested constructions or parsing inconsistencies in the AI-generated text. DLT_AVG remains relatively stable across corpora, with Web corpora showing slightly higher values (up to 0.418), indicative of longer dependencies.

Left Embedding (LE_AVG) and Nearest Neighbour Distance (NND_AVG) values are higher in human corpora, particularly Children and Web-UK, reflecting greater structural complexity and non-linear sentence organisation. In contrast, AI corpora demonstrate shallower embedding and less variation in dependency structure.

8.2.2 Dependency-Based Metrics: Maximum Values

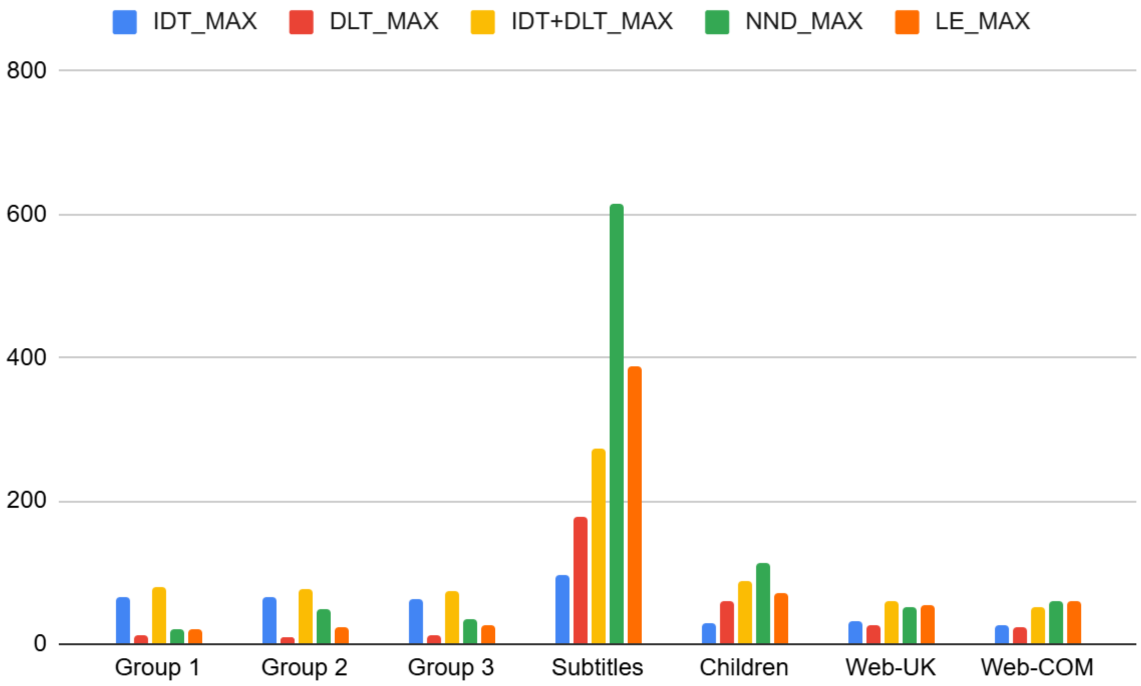

Figure 3. Dependency-Based Metrics: Maximum Values

Maximum values highlight outlier constructions and offer insight into the range of syntactic possibilities within each corpus. Subtitles exhibit extreme maxima across nearly all metrics, with NND_MAX reaching 614 and LE_MAX 389, likely due to informal constructions or parsing inconsistencies.

AI groups also produce relatively high maximum values (e.g., IDT+DLT_MAX of 79 in Group 1), indicating occasional deeper structures, but their extreme values remain below those of the Children and Web corpora. Web-UK and Web-COM maintain high LE_MAX and NND_MAX, signalling structurally complex constructions.

8.2.3 Dependency-Based Metrics: Sum Values

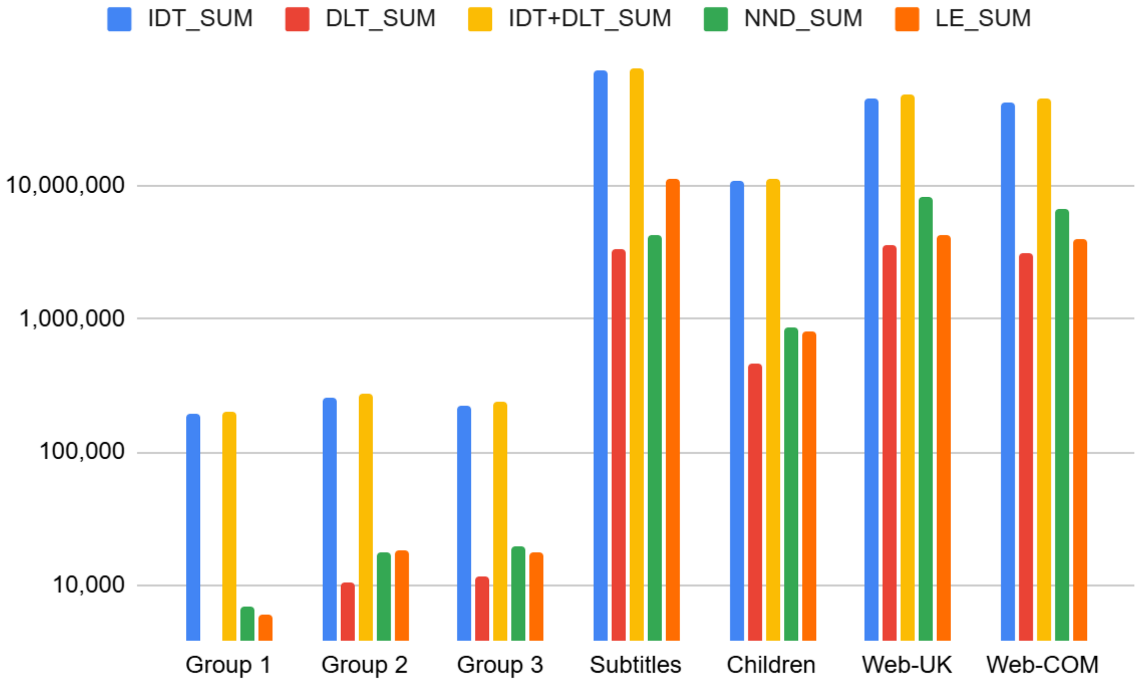

Figure 4. Dependency-Based Metrics: Sum Values

As expected, the sum values are highly influenced by corpus length. Subtitles, being the largest corpus, show extremely high totals across all metrics. Web corpora also demonstrate considerable cumulative syntactic activity, consistent with their larger size and content density.

AI corpora are smaller in volume, and while their sum values are meaningful internally, they are not directly comparable to human corpora without normalisation. For this reason, interpretation focuses primarily on average values to ensure fair comparison.

Chapter 9: Analysis and Result Discussion

This chapter presents a comprehensive and critical analysis of the syntactic complexity and syntactic processing complexity metrics applied across the AI-generated and human-authored corpora. The interpretation and contextualisation of the results includes both the comparison of the results in different prompt groups, reflections on the effectiveness of the prompt strategies applied, and implications of these results for the field of generative AI and computational linguistics research.

The discussion is organised according to two main metric categories: the first one presents clause-based syntactic complexity, dependency-based syntactic processing complexity metric results interpretation, as well as the results from the application of the emotion recognition tool; the second one describes more practical considerations and is further subdivided to highlight specific trends and cross-sectional insights.

9.1 Clause-Based Syntactic Complexity: Analysis

Clause-based syntactic complexity metrics provide information on how syntactic resources are distributed at the sentence and clause level; these features reflect linguistic choices related to subordination, coordination, nominal complexity, and clause structure.

The clause-based syntactic complexity metrics examined in this study (MLS, MLC, DC/C, CP/C, CN/C and C/S) offer an understanding of the structural complexity in AI-generated and human-authored corpora. The analysis of each of these metrics individually can uncover important patterns in how syntactic constructions differ across the datasets, highlighting the variation of these patterns in various types of texts.

Mean Length of Sentence (MLS)

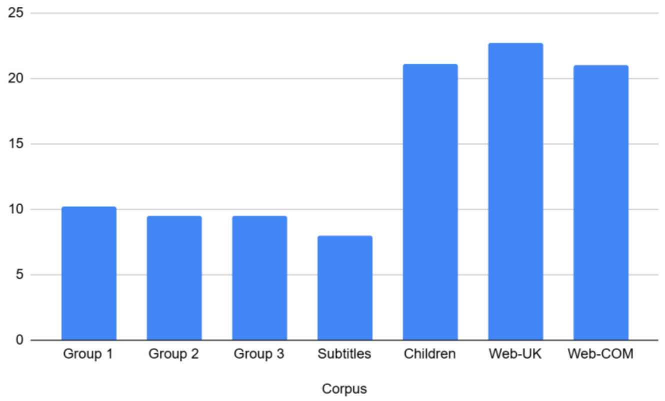

Figure 5. MLS results

The AI-generated groups (Groups 1–3) exhibit moderately long sentences, with Group 1 having the longest average sentence length among them. These values suggest that AI-generated texts tend to have more compact sentence construction. The lowest number of MLS was found in the Subtitles corpus, meaning that this corpus represents mainly informal spoken style textual constructions, which reflect the length of the sentence.

In contrast, human-written Children and Web corpora show substantially higher MLS values (more than 20), which are double those of the AI groups. This demonstrates a higher degree of elaboration and more extended syntactic structure in the texts that were meant to be written, but are natural in terms of their production.

Mean Length of Clause (MLC)

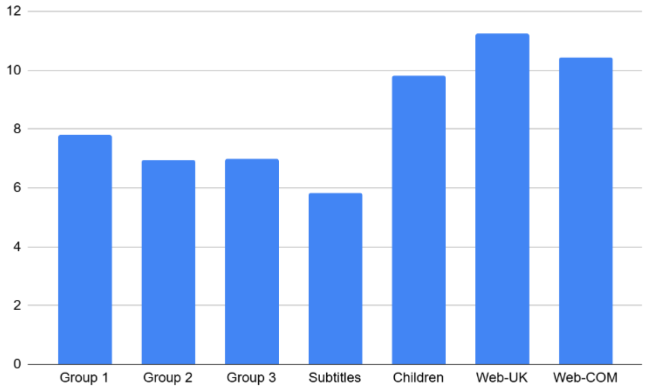

Figure 6. MLC results

It follows a similar trend to MLS. AI groups display shorter clauses, as well as the Subtitles corpus, reinforcing its simplified spoken structure. Human-authored corpora again diverge, particularly the Web-UK corpus, suggesting dense intra-clausal structures. The Children corpus also scores high on this metric, indicating that children's books, while designed for readability, still retain more complex internal clause structures than all the AI outputs.

Dependent Clauses per Clause (DC/C)

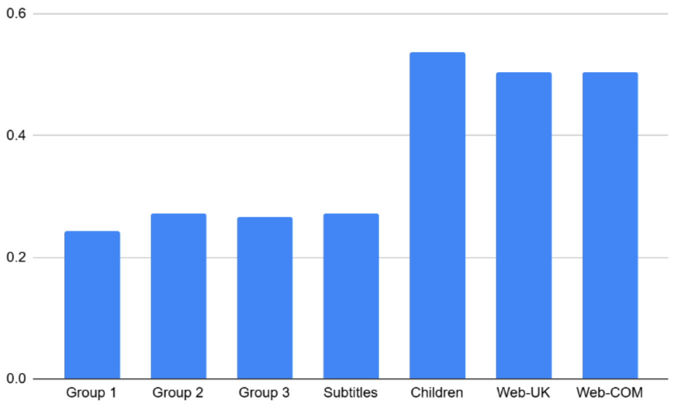

Figure 7. DC/C results

The data from this metric application offers insights into the subordination depth of the syntactic structures used across the corpora, revealing how frequently and extensively subordinate clauses are embedded within main clauses. Values for AI groups are relatively modest, indicating a limited use of embedded clauses. The Subtitles corpus aligns closely with this trend, also retraining the casual, informal style that usually does not include various subordinate contractions. In contrast, the Children corpus and Web corpora feature markedly more subordinate structures, pointing to higher levels of syntactic embedding in human-authored texts.

Coordinate Phrases per Clause (CP/C)

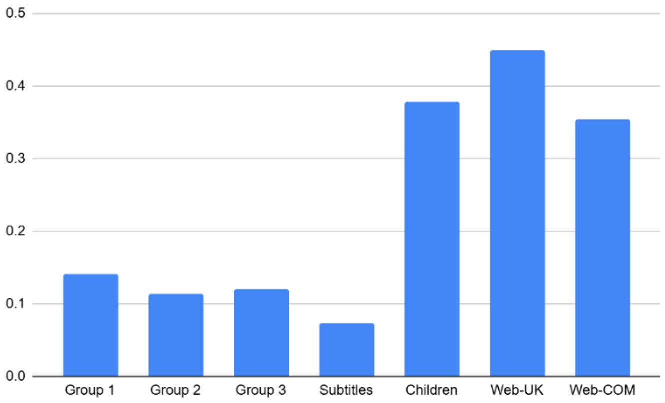

Figure 8. CP/C results

AI groups demonstrate restrained coordination, with Group 1 showing the highest CP/C among them; Subtitles exhibit even less coordination, reflecting a stylistic preference for simpler clause structures in dialogue. Human corpora again display greater syntactic richness, especially Web-UK and Children, suggesting more frequent use of coordination as a stylistic and structural device.

Complex Noun Phrases per Clause (CN/C)

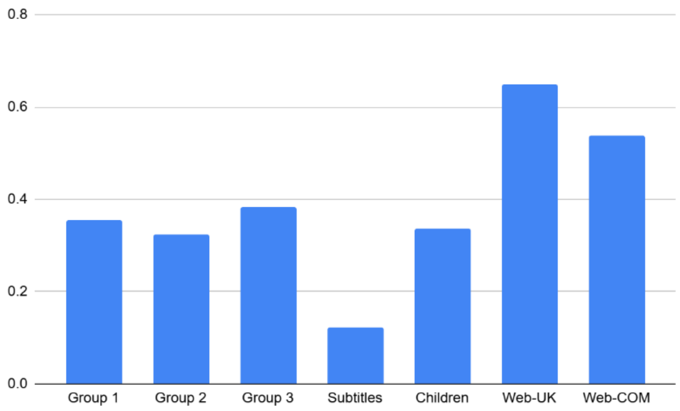

Figure 9. CN/C results

CN/C reveals differences in nominal complexity. Group 3 shows the highest CN/C among AI texts, suggesting that more intricate noun phrase structures emerged under more complex prompting conditions. Groups 1 and 2 yield slightly lower values, implying a more conservative use of complex nominals. Subtitles register the lowest CN/C value, again consistent with conversational tendencies. The Web corpora, particularly Web-UK, display a strong inclination toward structurally dense noun phrases.

Clauses per Sentence (C/S)

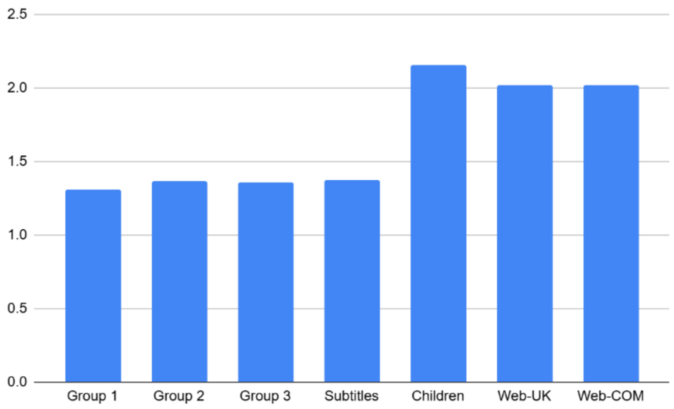

Figure 10. C/S results

C/S sheds light on overall sentence complexity. All corpora fall within a relatively narrow range. All of the AI texts maintain low C/S values, suggesting a preference for simpler sentence constructions, as well as the Subtitles corpus. In contrast, Children, Web-UK, and Web-COM show significantly higher values, indicating more layered syntactic organisation and a frequent use of embedded or coordinated clauses within single sentences.

In summary, the clause-based metrics show a clear difference in syntactic complexity between AI-generated and human-authored corpora. While AI groups demonstrate moderate syntactic complexity with slight differences across prompting styles (e.g. Group 1's higher MLS/MLC and Group 3's higher CN/C), they fall short of the structural depth exhibited by human-written texts, especially in subordination (DC/C), coordination (CP/C), and nominal density (CN/C). The Subtitles corpus shares some similarities with AI outputs, but exhibits even more simplified structures. Human texts, particularly those from the Web corpora, emerge as significantly more complex across all metrics, suggesting a richer and more variable syntactic repertoire.

Taken together, these metrics allow for a multifaceted comparison of syntactic behaviour across AI-generated and human-authored corpora. The interplay between subordination, coordination, and nominal complexity illustrates how different corpora balance clarity, formality, and density. While some AI outputs demonstrate attempts to replicate sophisticated syntactic styles, they may still fall short of the embeddedness and variability found in natural language, particularly in domains with high informational demands. The metric-specific divergences thus highlight both the current achievements and limitations of LLMs in mimicking human-like syntax.

9.2 Dependency-Based Syntactic Complexity: Analysis

The dependency-based syntactic complexity metrics examined in this study (IDT, DLT, IDT+DLT, NND, LE) offer a perspective on the cognitive load needed to process the sentences. These were computed using MAX and AVG values to reflect both the maximum and the mean syntactic processing load across corpora. While the numbers for SUM values were also reported for completeness in the previous chapter, they are not central to the interpretation in this study, as they reflect outlier complexity and are better suited as data for future research.

Incomplete Dependency Theory (IDT)

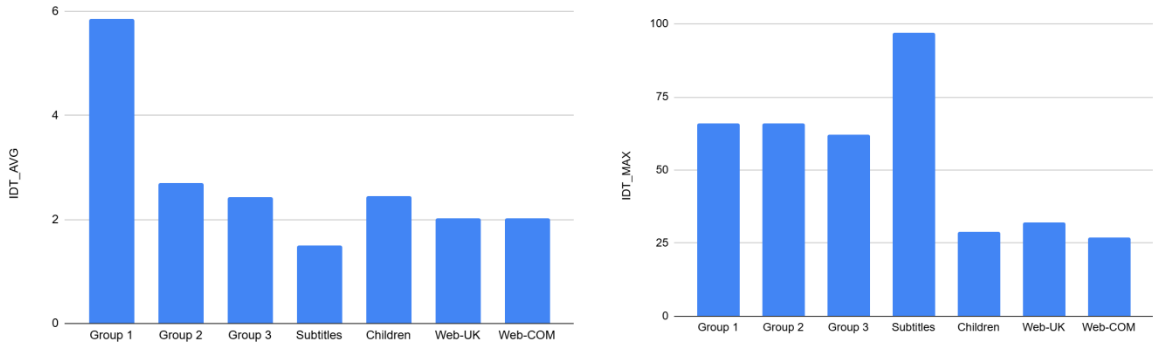

Figure 11. IDT results

The results for IDT reveal that human-written texts consistently exhibit a higher syntactic integration cost compared to AI-generated ones, both in terms of maximum (MAX) and average per sentence (AVG). This finding suggests that human authors tend to construct sentences with more complex syntactic structures. This aligns with the expectation that human texts exhibit greater structural variety and complex clause integration. In contrast, AI-generated texts appear more linear and exhibit a preference for shallow syntactic hierarchies, which reduces the integration demands and overall complexity.

Dependency Locality Theory (DLT)

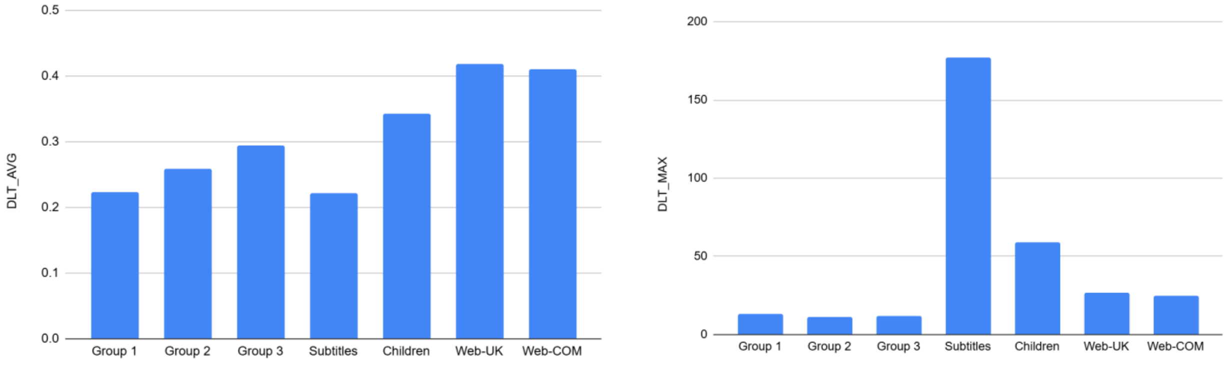

Figure 12. DLT results

The DLT values support the trend observed with IDT: human corpora show higher dependency locality costs, especially in the AVG results. This suggests that human authors more frequently create long-distance syntactic dependencies that require greater working memory resources to process. AI-generated texts, by comparison, demonstrate a tendency toward shorter dependency distances, likely as a strategy to maintain coherence and reduce the risk of error in longer constructions. This characteristic may stem from the generative nature of LLMs, which optimise for fluency and grammaticality but do not inherently simulate human memory constraints or communicative intentions that produce more varied syntactic spans.

Combined IDT + DLT

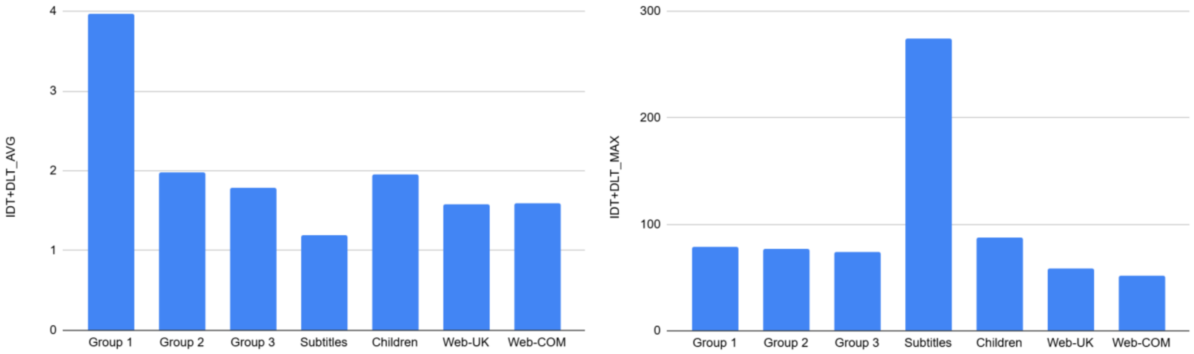

Figure 13. IDT+DLT results

The combined metric offers a more holistic perspective by aggregating both integration and locality costs. Once again, human corpora yield higher scores across MAX and AVG, reinforcing the conclusion that human-written language entails more demanding syntactic processing overall. The AVG values provide further nuance, showing that even at the sentence level, human writers are more inclined to deploy complex constructions.

Nested Noun Distance (NND)

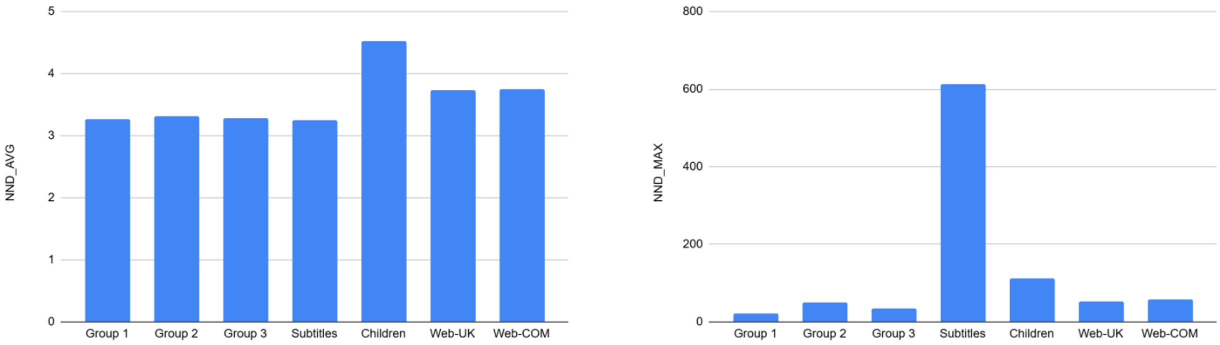

Figure 14. NND results

The NND metric, which reflects hierarchical embedding within sentences, shows one of the clearest differences between AI and human corpora. Human texts include more deeply nested structures, particularly in the AVG metric, indicating a frequent use of subordinate and embedded clauses. This aligns with the common linguistic observation that embedding is a hallmark of syntactic complexity and a common feature of naturalistic human discourse. In contrast, AI-generated content appears to avoid deep nesting, likely as a byproduct of autoregressive generation models, which favour low-risk and high-probability output. The relatively lower NND values in AI corpora suggest a stylistic preference for flat or right-branching structures that are easier to generate but syntactically simpler.

Left Embedding (LE)

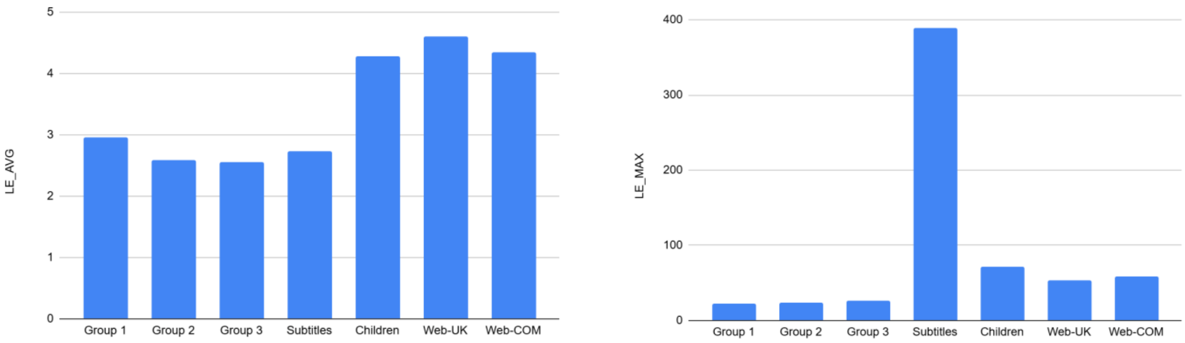

Figure 15. LE results

Left-embedding measures the extent to which subordinate clauses or complex structures occur early in the sentence, often increasing the processing burden due to delayed main clause resolution. The LE results show that human corpora include a greater number of such constructions, particularly in the AVG metric, again supporting the hypothesis that human language is more demanding in terms of real-time parsing. AI-generated texts show a notable avoidance of early embeddings, often postponing or avoiding them altogether in favour of clause-final elaboration. This again reflects optimisation strategies prioritised by language models: increasing clarity and grammaticality at the expense of syntactic depth.

Summary Interpretation

Across all the metrics, more complex human-authored texts consistently demonstrate greater syntactic complexity, both in total and average terms. These findings support the hypothesis that AI-generated language, while grammatically accurate and fluent, remains syntactically conservative. The absence of depth and nesting in AI outputs may point to a fundamental limitation in how LLMs simulate natural language syntax, particularly in narrative or creative domains where complexity contributes to nuance and style.

Overall, this analysis provides strong empirical evidence that syntactic complexity metrics can serve as a reliable comparison tool serve at differentiating AI and human-generated texts. It also highlights the current limitations of generative language models, particularly their inability to replicate the full syntactic richness of human communication, offering valuable insights for further developments in both AI modelling and corpus linguistics.

9.3 Effects of Prompt Complexity on Syntactic Complexity

This section evaluates the relationship between prompt complexity and the syntactic characteristics of the texts generated by ChatGPT. As it was already stated, the three experimental groups were designed with increasing levels of prompt specificity: Group 1 received the simplest and most general prompts, Group 2 was prompted using moderately elaborated instructions, and Group 3 employed the most detailed and controlled prompts. The working hypothesis was that simpler prompts would result in syntactically simpler texts, while more structured prompts would lead to richer and more sophisticated outputs.

However, the results challenge this initial assumption. When examining both clause-based and dependency-based metrics, a more nuanced interaction between prompt structure and linguistic output emerges.

Clause-based metrics provide a direct insight into sentence and clause length, subordination, and internal phrase complexity. Group 1, which was expected to produce the least complex texts, unexpectedly generated the longest sentences (Mean Length of Sentence = 10.179) and longest clauses (Mean Length of Clause = 7.789) among all three groups. Additionally, this group exhibited the highest rate of coordination per clause (CP/C = 0.141), suggesting that the model defaulted to producing syntactically dense but stylistically unvaried constructions when given minimal prompting. Conversely, measures more closely associated with linguistic sophistication, such as dependent clauses per clause (DC/C) and complex noun phrases per clause (CN/C), were highest in Groups 2 and 3. Specifically, Group 2 displayed the highest value for DC/C (0.272), while Group 3 led in CN/C (0.383), indicating a preference for greater syntactic embedding and nominal complexity in response to more detailed prompts.

The average number of clauses per sentence (C/S) was also higher in Groups 2 (1.368) and 3 (1.361) than in Group 1 (1.307), suggesting a greater degree of clause chaining and information layering in the texts generated with more complex prompting. This further supports the conclusion that greater prompt complexity prompts ChatGPT to produce more hierarchically structured discourse units.

Turning to dependency-based syntactic metrics, which assess cognitive load and syntactic integration during processing, the differences between groups become even more pronounced. Group 1 texts exhibited the highest average syntactic integration cost (IDT_AVG = 5.861) and the highest combined integration and distance load (IDT+DLT_AVG = 3.966). These values are significantly elevated in comparison to Group 2 (IDT_AVG = 2.707; IDT+DLT_AVG = 1.976) and Group 3 (IDT_AVG = 2.428; IDT+DLT_AVG = 1.791). This finding is especially significant, as it suggests that texts produced from the simplest prompts imposed a greater cognitive load in terms of dependency resolution, despite lacking deliberately crafted syntactic sophistication.

Furthermore, Dependency Locality Theory (DLT) values, which measure the linear distance between dependent elements and their heads, were highest in Group 3 (DLT_AVG = 0.295), suggesting that although these texts did not have the highest integration cost, they involved more long-distance dependencies. This aligns with the expectation that more specific prompts elicit structured planning from the model, leading to more syntactically distributed constructions. Likewise, Group 3's output featured the highest maximum values for syntactic distance (DLT_MAX = 12) and complex noun dependencies, reinforcing the notion that detailed prompt instructions encourage more sophisticated structural planning.

Across all groups, the values for Nested Nouns Distance (NND) and Left Embeddedness (LE) were relatively stable, although Group 2 displayed a slightly elevated average NND (3.309), and Group 1 exhibited the highest average LE (2.963). This again suggests that without structured prompts, the model may produce heavily embedded constructions in an uncontrolled fashion, contributing to overall cognitive load but not necessarily increasing stylistic or grammatical sophistication.

Taken together, these findings suggest a counterintuitive pattern. Rather than producing the most structurally simple texts, the least specific prompts resulted in outputs with higher integration costs, longer clauses and sentences, and a higher frequency of coordinated structures. This indicates a tendency for the model to default to verbose, formulaic, and syntactically heavy constructions when insufficiently guided. On the other hand, more elaborate prompts, particularly in Group 3, led to more deliberate and structured outputs, evidenced by higher levels of nominal complexity, increased use of subordination, and a more balanced distribution of syntactic load.

This outcome reflects the model's sensitivity to prompt structure: when provided with minimal instruction, ChatGPT appears to fall back on generic discourse strategies, often resulting in dense but stylistically flat outputs. When guided by more detailed prompts, the model demonstrates a greater ability to replicate human-like sentence planning, producing output with more structural variety and a more balanced distribution of syntactic complexity.

These results highlight the central role of prompt design in shaping the syntactic nature of AI-generated text. They indicate that detailed, thoughtfully constructed prompts do not merely improve content relevance but also facilitate the generation of linguistically sophisticated output. This has implications for prompt engineering in both applied NLP and linguistic research, where the control of syntactic features is often critical.

In summary, while it was initially hypothesised that prompt simplicity would lead to simpler outputs, the analysis demonstrates the opposite. Minimal prompts generated output with high dependency load and coordination, whereas the use of increasingly elaborated prompts resulted in more syntactically balanced and nuanced texts. Thus, prompt complexity is not merely a matter of semantics or task relevance; it also governs the structural depth and processing characteristics of the language produced by the model.

9.4 Comparative Analysis of Emotional Variation in Prompted Outputs

The analysis of emotional content in the generated texts revealed notable patterns across all three prompt conditions. Group 1, which was generated using the simplest prompts and produced shorter texts overall, exhibited relatively low emotional density. The most frequently occurring emotion in this group was Joy (148 instances), followed closely by Anger (134) and Fear (133). The remaining emotional categories, Disgust (89), Surprise (88), and Sadness (70), were present in lower numbers. These modest totals likely reflect the shorter average length of Group 1 texts, which limited the expressive space available for emotional variation.

By contrast, Groups 2 and 3, which were generated using more elaborated and highly structured prompts, respectively, produced outputs of comparable length, making their emotional content directly comparable in both frequency and diversity. In Group 2, Joy was the dominant emotion (403), with Fear (395), Anger (347), and Disgust (346) closely following. Surprise (248) and Sadness (158) were also substantially represented. This relatively balanced distribution across both positive and negative emotions suggests that moderately complex prompts elicited emotionally rich and varied language from the model. Notably, while Joy remained the most common emotion, the high presence of Fear and Disgust indicates the model's ability to generate affectively layered output even under neutral or narrative prompt conditions.

In Group 3, which involved the most complex and directive prompts, the emotional profile shifted markedly. Here, Anger emerged as the most frequent emotion (476), with Fear close behind (474). Disgust (389) and Joy (375) also featured prominently, while Sadness (229) and Surprise (210) maintained relatively high frequencies. Compared to Group 2, Group 3 displayed a greater concentration of negative emotions, suggesting that detailed prompts may encourage the model to construct more emotionally charged or conflict-driven content. This shift could be attributed to prompt structures that incorporated narrative complexity, interpersonal dynamics, or explicit emotional framing, all of which are likely to trigger affective language choices within the model’s generative process.

The contrast between Groups 2 and 3 is particularly revealing. Despite having similar output lengths, Group 2's emotional spectrum is more evenly distributed across positive and negative affect, while Group 3 shows a clear skew toward high-arousal negative emotions, especially Anger and Fear. This suggests that increasing prompt complexity does not merely enhance emotional richness but may also influence the valence and intensity of the generated affect. In essence, while Group 2 reflects a more balanced emotional register, Group 3 reveals a tendency toward dramatic, emotionally heightened narratives, potentially reflecting the model’s learned association between complexity and affective intensity.

These findings underscore the model's capacity to modulate emotional content in response to prompt design. When controlling for text length, as in the comparison between Groups 2 and 3, it becomes clear that prompt complexity plays a significant role not only in shaping syntactic structure but also in steering the expressive and stylistic tone of the output. This highlights the importance of carefully crafted prompt engineering when emotional framing or expressive variation is desired in AI-generated text.

9.5 Theoretical and Practical Implications

The results of this study reveal critical insights into how prompt design influences not only the syntactic complexity of AI-generated language but also its emotional expressiveness. These findings carry wide-ranging implications across computational linguistics, language pedagogy, psycholinguistics, and the broader development and implications of generative AI in various fields.

The behaviour of LLMs is inherently shaped by both the data on which they are trained and the structural design of the model itself. When examining syntactic tendencies, it becomes apparent that the model often favours simpler, more conventional sentence constructions. This inclination likely mirrors the nature of its underlying corpora, which may contain a disproportionate amount of generalist or easily processed text. However, the exact composition of these training datasets is largely undisclosed, leaving researchers without the ability to conduct a detailed audit of potential sources of bias. As a result, understanding why the model prefers certain syntactic patterns over others remains challenging. Beyond simple syntactic preferences, such biases may affect more subtle aspects of language production, including the choice of complex noun phrases, clause coordination, or discourse-level cohesion. This opacity complicates both the interpretation of model output and the identification of potential limitations when applying LLMs to tasks requiring linguistic precision or stylistic diversity.

9.5.1 Cognitive Load, Readability, and Syntactic Accessibility

From a psycholinguistic perspective, these findings highlight one of the challenges in natural language generation: the necessity to balance between fluency and processing efficiency and syntactic richness and cognitive engagement. While AI-generated texts may rather choose structures and language that are easier to parse and process and that are more accessible for general readers, and particularly for those with lower proficiency, this reduction in complexity may result in a loss of logical depth and discourse cohesion. For tasks demanding high precision or elaborate argumentation, such as legal writing or academic discourse, this syntactic simplification may affect the clarity and structure typical of these types of texts, potentially leading to misunderstandings or insufficiently nuanced communication.

9.5.2 Implications for Prompt Engineering and NLG Design

The findings reinforce the central role of prompt design in modulating AI behaviour. Although increased prompt complexity encouraged marginal gains in syntactic elaboration and stylistic diversity, these changes were not always proportional. The observed ceiling effect, where overly complex prompts failed to produce correspondingly rich output, suggests intrinsic constraints in current LLM architectures.

Consequently, relying solely on prompt manipulation may not suffice to generate outputs comparable in complexity to expert human writing. More effective solutions may involve hybrid approaches: combining prompt engineering with fine-tuning on genre-specific corpora, using reinforcement learning to reward structural richness, or incorporating explicit syntactic control mechanisms during generation. Such innovations could help bridge the current gap between surface fluency and deeper linguistic organisation.

9.5.4 Applications in Language Education and Communication

The relative syntactic simplicity and emotional accessibility of AI-generated texts position them as potentially valuable tools in language learning, particularly at the beginner to intermediate levels. Their reduced processing load makes them ideal for comprehensible input, supporting vocabulary acquisition and early syntactic pattern recognition.

However, their usefulness diminishes in advanced pedagogical contexts that demand exposure to authentic syntactic variation and expressive nuance. Learners at higher proficiency levels require more than surface fluency; they need access to the kinds of subordinated, embedded, and rhetorically marked structures that characterise expert writing. AI writing tools tailored to dynamically adapt syntactic complexity to learner profiles could address this gap, scaffolding linguistic development in a more personalised and effective way.

9.5.5 Cautions for Corpus Construction and Linguistic Research

The study highlights key limitations when incorporating AI-generated language into linguistic corpora. While such texts may exhibit lexical variety and cohesion, their tendency to avoid complex syntactic and rhetorical structures could skew linguistic analyses, particularly those related to sentence planning, subordination, or discourse organisation.

Researchers must be cautious when using AI-generated corpora in computational linguistics or psycholinguistic studies. Validation against naturalistic human-written texts is essential to avoid introducing structural biases that may affect models of language complexity, comprehension, or acquisition. Nevertheless, AI-generated corpora can still serve as controlled environments for studying stylistic effects, prompt variation, or text generation mechanics, provided their limitations are acknowledged.

9.5.6 Broader Perspectives on AI and Human-Like Communication

Ultimately, the findings underscore the current limits of AI in emulating the full expressive and structural range of human language. Despite advancements in large-scale training and transformer architectures, LLMs still struggle to replicate the layered syntactic planning and emotional modulation found in skilled human writing. Their outputs tend to favour fluency over depth, clarity over nuance, and simplicity over complexity.

This invites continued interdisciplinary collaboration between AI researchers, linguists, educators, and cognitive scientists to refine both the theoretical understanding and practical design of natural language generation systems. Whether for creative writing, persuasive discourse, or technical documentation, advancing AI's capacity for true human-like expressivity will depend on confronting and addressing these syntactic and stylistic limitations.

Chapter 10: Challenges, Limitations, and Future Directions of the Study

This chapter addresses the key challenges encountered during the study, acknowledges the inherent limitations affecting the findings, and outlines promising avenues for future research. These reflections are critical for situating the study within the broader field of computational linguistics and AI-generated language research, ensuring transparency, and guiding subsequent investigations.

10.1 Challenges Encountered in the Study

10.1.1 Data Collection and Corpus Construction

One of the primary challenges was the compilation of comparable corpora for meaningful syntactic complexity analysis. Aligning AI-generated texts with human-authored corpora across multiple genres (subtitles, children’s stories, web texts) required careful balancing of length, thematic content, and style to ensure comparability.

Generating sufficiently large and representative AI corpora using prompt-based methods posed practical difficulties. While the OpenAI API facilitated text generation, prompt design had to be meticulously crafted and iteratively refined to elicit outputs of varying syntactic complexity. However, inherent model constraints limited the diversity and depth of structures that could be produced, even with complex prompts.

Moreover, the constraints of the OpenAI API posed a logistical limitation during generation. The API functions as a stateless system, which means prompts must be finalised before execution and cannot be modified dynamically mid-generation. This makes iterative refinement of prompts in real time infeasible: if a prompt yields suboptimal or overly simplistic output, the only recourse is to redesign and resubmit it in a new session—an approach that can quickly become time-consuming and token-expensive. Therefore, extensive pretesting of all prompts was necessary to minimise waste and ensure that generated outputs aligned with the study’s syntactic and stylistic targets before scaling up to corpus-level production.

These intertwined challenges underscore the difficulty of balancing methodological consistency, computational feasibility, and resource constraints when working across both human-authored and AI-generated texts. Careful prevalidation of prompts, strategic batching of large corpora, and error-aware preprocessing were essential to maintaining the reliability and comparability of the final datasets.

10.1.2 Syntactic Parsing and Metric Computation

Another significant challenge emerged from the scale and complexity of the reference corpora used in this study. Resources such as the full OpenSubtitles corpus and the Leipzig Web Corpus are extensive in size, often comprising millions of words. Processing such large datasets imposes high computational demands on both memory and processing time, particularly during the syntactic parsing phase. The parsing of these corpora had to be conducted in batches, with intermediate error handling and output verification, to mitigate memory overflows and parsing inconsistencies. Moreover, parsing tools like spaCy, while effective for most textual inputs, can struggle with the stylistic fragmentation and informal register of subtitles or user-generated web content, necessitating extensive pre-processing to clean and segment the data adequately for analysis.

Moreover, the choice of syntactic complexity metrics, while grounded in established literature, is not exhaustive. Some nuanced syntactic phenomena, such as discourse-level dependencies or pragmatic markers, are not captured by the selected measures, limiting the scope of structural analysis.

10.1.3 Scope of Prompt Engineering

The study focused on three groups of prompts with varying complexity to probe their effects on syntactic output. However, this scope may not fully represent the vast space of possible prompt designs, styles, or instructions. More sophisticated or hybrid prompting strategies could produce richer syntactic variation, but were beyond the study's timeframe and resources.

10.1.4 Cross-Genre and Register Variation

While genre diversity strengthens the study, it also complicates direct comparisons. Subtitles, children's stories, and web corpora differ inherently in communicative purpose, audience, and stylistic conventions, influencing syntactic complexity independently of authorship (human vs AI). Isolating the effect of authorship from genre effects remains challenging.

10.2 Future Directions for Research

10.2.1 Expanding Prompt Engineering and Control Mechanisms

Future studies should explore a wider variety of prompt formulations, including hierarchical, multi-step, or constraint-based prompts to push syntactic complexity boundaries. Integrating controllable generation techniques that explicitly target syntactic features could enable more nuanced output.

10.2.2 Incorporating Multilingual and Cross-Linguistic Perspectives

Extending analysis to other languages and multilingual models would provide insights into the universality or language-specificity of AI syntactic behaviour. Different languages with varying syntactic typologies may challenge or reveal different model limitations.

10.2.3 Improving Parsing and Annotation Tools for AI-Generated Texts

Development of syntactic parsers and annotation schemes optimised for AI-generated or genre-diverse texts could reduce parsing errors and improve metric reliability. Tailored tools might better capture emerging patterns or idiosyncrasies of machine-generated language.

10.2.4 Longitudinal and Interactive Generation Studies

Research could investigate how iterative or interactive prompting influences syntactic complexity, including feedback loops where humans refine AI output. Longitudinal studies tracking model improvements over versions would also clarify how syntactic sophistication evolves.

10.2.5 Evaluating the Impact of Syntactic Complexity on User Comprehension and Engagement

Empirical studies involving human readers to assess how differences in AI text complexity affect comprehension, memory, or engagement would ground computational findings in real-world relevance. This user-centred approach can guide AI text design for educational, professional, or creative uses.

Conclusion

This thesis has investigated the linguistic properties of AI-generated texts, with a particular focus on syntactic complexity and distribution of emotions in these texts, and evaluates these features compared with human-authored corpora. By constructing three AI-generated corpora with varying prompt complexity and comparing them with three human reference corpora, the study has provided evidence on both the potential and the limitations of LLM outputs in corpus linguistics.

The first major finding concerns syntactic complexity. The analysis revealed that across all metrics, AI-generated texts displayed consistent grammaticality and fluency, but with lower variability than advanced human corpora: sentences tended to be shorter, clauses structurally simpler, and constructions more predictable, which resembles the data obtained from the corpus of film subtitles; however, the generated texts did not exhibit enough structural complexity to match the complexity of the Children books corpus or web corpus. Dependency-based measures further revealed that AI texts prioritise efficient structures, resulting in lower integration costs and reduced locality effects compared to human-authored texts. While increased prompt complexity encouraged greater use of subordination and coordination, the outputs never fully matched the diversity observed in natural corpora. This suggests that LLMs, while capable of producing sophisticated syntax under specific conditions, default to structural regularity unless explicitly guided.

The second key finding relates to sentiment analysis: AI-generated corpora demonstrated a strong bias toward neutral and mildly positive affect, producing texts that lack negative emotions, limiting the pragmatic and stylistic richness of the outputs.

The third major finding is methodological: prompt complexity significantly shapes output quality. Simple prompts produced highly regular and limited texts, whereas complex prompts elicited outputs closer to human-like variation. This underscores the importance of prompt design in both research and practice. However, even the most carefully constructed prompts could not fully compensate for structural and pragmatic limitations embedded in the models.

Taken together, these findings contribute to computational linguistics in three key ways. First, they provide a detailed corpus-based comparison of AI and human texts, focusing on syntactic depth and emotional distribution. Second, they highlight the methodological significance of prompt engineering, demonstrating its role in shaping corpus-level features of

AI outputs. Third, they raise broader theoretical and ethical questions about the use of AI-generated corpora, particularly in contexts where authenticity and representativeness are essential.

It should also be noted that the study is not without limitations. The focus on English excludes insights from multilingual and low-resource contexts, where LLMs often underperform. The size of the AI-generated corpora, while systematically sampled, remains smaller than many large-scale human corpora. Moreover, the analysis centred on syntax and sentiment, leaving other dimensions, such as discourse coherence, figurative language, and multimodality, for future exploration.

In conclusion, this thesis demonstrates that LLM-generated corpora occupy a complex position between approximation and divergence. They are capable of producing fluent, coherent, and contextually relevant text, particularly when guided by carefully engineered prompts. Yet, they remain distinguishable from human-authored language in their structural regularity and emotional limitations. For linguistics, this offers both opportunity and caution: AI-generated corpora can serve as valuable experimental tools, but they cannot substitute for the richness and variability of authentic human language. The boundary between human and machine authorship is narrowing, but it is not yet erased.

References

Abbas, F., & Taeihagh, A. (2024). Unmasking deepfakes: A systematic review of deepfake detection and generation techniques using artificial intelligence. *Expert Systems with Applications, 252*(Part B), 124260. https://doi.org/10.1016/j.eswa.2024.124260

Ali, I., Atuhurra, J., Kamigaito, H., & Watanabe, T. (2025). *HLU: Human vs LLM-generated text detection dataset for Urdu at multiple granularities*. In *Proceedings of the 31st International Conference on Computational Linguistics* (pp. 3495–3510). Association for Computational Linguistics. https://aclanthology.org/2025.coling-main.235/ACL Anthology+1ACL Anthology+1

Anam, R. K. (2025). *Prompt engineering and the effectiveness of large language models in enhancing human productivity*. arXiv. https://doi.org/10.48550/arXiv.2507.18638

Ban, Y., Wang, R., Zhou, T., Cheng, M., Gong, B., & Hsieh, C.-J. (2024). *Understanding the impact of negative prompts: When and how do they take effect?* arXiv. https://doi.org/10.48550/arXiv.2406.02965

Banerjee, S., & Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. *In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* (pp. 65–72). Association for Computational Linguistics. https://aclanthology.org/W05-0909/

Bang, Y., Zhang, S., Madaan, A., et al. (2023). *Multitask Prompted Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. arXiv preprint arXiv:2302.04023.*

Bansal, R., Saini, S., & Pruthi, N. (2023). Promoting customer engagement through artificial intelligence: A systematic literature review. *Academy of Marketing Studies Journal, 27*(S5), 1–8. https://www.researchgate.net/publication/377658640_Promoting_customer_engagement_through_artificial_intelligence_-a_systematic_literature_review

Bao, W., Cao, Y., Yang, Y., Che, H., Huang, J., & Wen, S. (2025). Data-driven stock forecasting models based on neural networks: A review. *Information Fusion, 113,* 102616. https://doi.org/10.1016/j.inffus.2024.102616

Bardol, K. (2025). ChatGPT reads your tone and responds accordingly — until it does not: Emotional framing induces bias in LLM outputs. arXiv. https://doi.org/10.48550/arXiv.2507.21083

Barzilay, R., & Lapata, M. (2008). Modeling local coherence: An entity-based approach. *Computational Linguistics, 34*(1), 1–34. https://doi.org/10.1162/coli.2008.34.1.1

Bayram, F., Ahmed, B. S., & Kassler, A. (2022, March 21). *From concept drift to model degradation: An overview on performance-aware drift detectors* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2203.11070

Berber Sardinha, T. (2024). *AI-generated vs human-authored texts: A multidimensional comparison*. *Applied Corpus Linguistics, 4*(1), 100083. https://doi.org/10.1016/j.acorp.2023.100083

Bestgen, Y. (2024). Measuring lexical diversity in texts: The twofold length problem. *Language Learning, 74*(3), 638–671. https://doi.org/10.1111/lang.12630

Bhaila, K., Van, M.-H., & Wu, X. (2024). *Soft prompting for unlearning in large language models*. *arXiv*. https://doi.org/10.48550/arXiv.2406.12038

Bhandari, P., Fay, N., Wise, M., Datta, A., Meek, S., Naseem, U., & Nasim, M. (2025, February 17). *Can LLM agents maintain a persona in discourse?* arXiv. https://doi.org/10.48550/arXiv.2502.11843

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media.

Blevins, T., Schmalwieser, S., & Roth, B. (2025). Do language models accommodate their users? A study of linguistic convergence. *arXiv*. https://doi.org/10.48550/arXiv.2508.03276

Bolukbasi, T., Chang, K. W., Zou, J. Y., Saligrama, V., & Black, M. (2016). *Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings.* arXiv preprint arXiv:1607.06520.

Breve, B., Cimino, G., & Deufemia, V. (2024). Hybrid prompt learning for generating justifications of security risks in automation rules. *ACM Transactions on Intelligent Systems and Technology, 15*(5), Article 103, 1–26. https://doi.org/10.1145/3675401

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2020). *Language models are few-shot learners*. Advances in Neural Information Processing Systems, 33, 1877–1901. https://arxiv.org/abs/2005.14165

Cabanac, G., Labbé, C., & Magazinov, A. (2021). *Tortured phrases: A dubious writing style emerging in science. Evidence of critical issues affecting established journals*. arXiv. https://arxiv.org/abs/2107.06751

Cañizares-Díaz, H., Piad-Morffis, A., Estevez-Velarde, S., Gutiérrez, Y., Almeida Cruz, Y., Montoyo, A., & Muñoz-Guillena, R. (2021). Active learning for assisted corpus construction: A case study in knowledge discovery from biomedical text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)* (pp. 216–225). INCOMA Ltd. https://aclanthology.org/2021.ranlp-1.26

Chaganty, A. T., Mussmann, S., & Liang, P. (2018). *The price of debiasing automatic metrics in natural language evaluation*. arXiv. https://doi.org/10.48550/arXiv.1807.02202

Chambers, F. (1997). What do we mean by fluency? *System, 25*(4), 535–544. https://doi.org/10.1016/S0346-251X(97)00046-8

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2023). *A survey on evaluation of large language models* (Version 9). arXiv. https://doi.org/10.48550/arXiv.2307.03109

Chen, B., Zhang, Z., Langrené, N., & Zhu, S. (2025). Unleashing the potential of prompt engineering for large language models. *Patterns, 6*(6), 101260. https://doi.org/10.1016/j.patter.2025.101260

Chen, H. W., & Istead, L. (2024). "Imagine a dress": Exploring the case of task-specific prompt assistants for text-to-image AI tools. *GI '24: Proceedings of the 50th Graphics Interface Conference*, Article No. 13, 1–8. https://doi.org/10.1145/3670947.3670972

Cheng, S. (2025). When journalism meets AI: Risk or opportunity? *Digital Government: Research and Practice, 6*(1), Article 12, 1–12. https://doi.org/10.1145/3665897

Cheung, H., & Kemper, S. (1992). Competing complexity metrics and adults' production of complex sentences. Applied Psycholinguistics, 13(1), 53-76. https://doi.org/10.1017/S0142716400005427

Chudleigh, S. (2024). *What is AI prompt chaining?* Botpress. Retrieved February 10, 2025, from https://botpress.com/blog/what-is-ai-prompt-chaining

Cornelius, J., Lithgow-Serrano, O., Mitrovic, S., Dolamic, L., & Rinaldi, F. (2024). *BUST: Benchmark for the evaluation of detectors of LLM-generated text*. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (pp. 8029–8057). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.444ACL Anthology+1ACL Anthology+1

Curry, N., Baker, P., & Brookes, G. (2023). Generative AI for corpus approaches to discourse studies: A critical evaluation of ChatGPT. *Journal of Corpora and Discourse Studies, 6*(1), 100082. https://doi.org/10.1016/j.acorp.2023.100082

Das, S., Tariq, A., Santos, T., Kantareddy, S. S., & Banerjee, I. (2023). Recurrent neural networks (RNNs): Architectures, training tricks, and introduction to influential research. In O. Colliot (Ed.), *Machine learning for brain disorders* (Chapter 4). Humana. https://www.ncbi.nlm.nih.gov/books/NBK597502/?utm_source=chatgpt.com

De la Iglesia, I., Goenaga, I., Ramirez-Romero, J., Villa-Gonzalez, J. M., Goikoetxea, J., & Barrena, A. (2025). *Ranking over scoring: Towards reliable and robust automated evaluation of LLM-generated medical explanatory arguments*. In *Proceedings of the 31st International Conference on Computational Linguistics* (pp. 9456–9471). Association for Computational Linguistics. https://aclanthology.org/2025.coling-main.634/ACL Anthology+1arXiv+1

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1810.04805

Doshi, A. R., & Hauser, O. P. (2024). Generative AI enhances individual creativity but reduces the collective diversity of novel content. *Science Advances, 10*(28), eadn5290. https://doi.org/10.1126/sciadv.adn5290

Dulaney, C. L., Davlin-Pater, C., & Cagle, J. A. B. (2023). Academic Performance: Prompting Strategic Selection of Resource Use. *College Teaching, 73(1),* 46–54. https://doi.org/10.1080/87567555.2023.2226383

Durmus, E., He, H., & Diab, M. (2020). FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5055–5070. https://doi.org/10.18653/v1/2020.acl-main.454

Eden, B. D. (2021). Children Stories Text Corpus (Version 1) [Data set]. Kaggle. https://www.kaggle.com/datasets/edenbd/children-stories-text-corpus

Ein-Dor, L., Shnarch, E., Dankin, L., Halfon, A., Sznajder, B., Gera, A., Alzate, C., Gleize, M., Choshen, L., Hou, Y., Bilu, Y., Aharonov, R., & Slonim, N. (2019). *Corpus wide argument mining – A working solution*. arXiv. https://doi.org/10.48550/arXiv.1911.10763

Elkhatat, A. M., Elsaid, K., & Almeer, S. (2023). Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text. *International Journal for Educational Integrity, 19*(17). https://doi.org/10.1007/s40979-023-00140-5

European Commission. (2019). *Ethics guidelines for trustworthy AI.* European Commission. Retrieved February 17, 2025, from https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

Explosion. (n.d.). spaCy: Industrial-strength natural language processing in Python. https://spacy.io

Fedoriv, Y., Pirozhenko, I., & Shuhaï, A. (2023). *Linguistic analysis of human- and AI-created content in academic discourse*. Journal of Vasyl Stefanyk Precarpathian National University. Philology, 10, 47–67. https://doi.org/10.15330/jpnuphil.10.47-67

Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative AI. *Business & Information Systems Engineering, 66*(2), 111–126. https://doi.org/10.1007/s12599-023-00834-7

Franceschelli, G., & Musolesi, M. (2024). *Creativity and machine learning: A survey*. ACM Computing Surveys, 56(11), Article 283. https://doi.org/10.48550/arXiv.2104.02726

Gibson, E. (2000). The dependency locality theory: A distance-based theory of linguistic complexity. In Image, language, brain: Papers from the first Mind Articulation Project Symposium (pp. 95–126). MIT Press.

Goldhahn, D., Eckart, T., & Quasthoff, U. (2012). Building large monolingual dictionaries at the Leipzig Corpora Collection: From 100 to 200 languages. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12) (pp. 759–765). European Language Resources Association (ELRA). https://wortschatz.uni-leipzig.de/en

Goyal, R., Kumar, P., & Singh, V. P. (2023). A systematic survey on automated text generation tools and techniques: Application, evaluation, and challenges. *Multimedia Tools and Applications, 82,* 43089–43144. https://doi.org/10.1007/s11042-023-15224-0

Gries, S. T. (2025). Corpus linguistics: Quantitative methods. In C. A. Chapelle (Ed.), *The encyclopedia of applied linguistics*. Wiley. https://doi.org/10.1002/9781405198431.wbeal20003

GSDC. (2024, October 24). *Top prompt engineering challenges and their solutions*. GSD Council. Retrieved February 12, 2025, from https://www.gsdcouncil.org/blogs/top-prompt-engineering-challenges-and-their-solutions?utm_source=chatgpt.com

Gstrein, O. J., & Beaulieu, A. (2022). How to protect privacy in a datafied society? A presentation of multiple legal and conceptual approaches. *Philosophy & Technology, 35*(1), 3. https://doi.org/10.1007/s13347-022-00497-4

Guan, J., Mao, X., Fan, C., Liu, Z., Ding, W., & Huang, M. (2021). Long text generation by modeling sentence-level and discourse-level coherence. *arXiv*. https://arxiv.org/abs/2105.08963

Guo, Y., Guo, M., Su, J., Yang, Z., Zhu, M., Li, H., Qiu, M., & Liu, S. S. (2024). *Bias in large language models: Origin, evaluation, and mitigation* (arXiv:2411.10915v1). arXiv. https://arxiv.org/abs/2411.10915v1

Guo, Z., Jin, R., Liu, C., Huang, Y., Shi, D., Supryadi, Yu, L., Liu, Y., Li, J., Xiong, B., & Xiong, D. (2023). *Evaluating large language models: A comprehensive survey*. arXiv. https://doi.org/10.48550/arXiv.2310.19736

Guriță, A.-E. (2025). SAID: A Social Media AI-generated Interface Dataset Using Prompt Engineering Methods Focused On Accessibility. *Proceedings of the International AAAI Conference on Web and Social Media*, *19*(1), 2446-2453. https://doi.org/10.1609/icwsm.v19i1.35947

Habib, S., Vogel, T., Anli, X., & Thorne, E. (2024). How does generative artificial intelligence impact student creativity? *Journal of Creativity, 34*(1), 100072. https://doi.org/10.1016/j.yjoc.2023.100072

Hanselowski, A., PVS, H., Schiller, B., Caspelherr, F., & Gurevych, I. (2019). A richly annotated corpus for different tasks in automated fact-checking. *arXiv*. https://arxiv.org/abs/1911.01214

Hariharan, M. (2024). *Semantic mastery: Enhancing LLMs with advanced natural language understanding*. arXiv. https://arxiv.org/abs/2504.00409

Hartmann, J. (2022). Emotion English DistilRoBERTa-base [Model]. Hugging Face. https://huggingface.co/j-hartmann/emotion-english-distilroberta-base/

Herbold, S., Hautli-Janisz, A., Heuer, U., Kikteva, Z., & Trautsch, A. (2023). *AI, write an essay for me: A large-scale comparison of human-written versus ChatGPT-generated essays. Scientific Reports, 13*(18617). https://doi.org/10.1038/s41598-023-45644-9

Hewamalage, H., Bergmeir, C., & Bandara, K. (2020). *Recurrent neural networks for time series forecasting: Current status and future directions. International Journal of Forecasting, 37*(1), 388–427. https://doi.org/10.1016/j.ijforecast.2020.06.008

Hinchman, K. A., & Moore, D. W. (2013). Close reading: A cautionary interpretation. *Journal of Adolescent & Adult Literacy, 56*(6), 440–509. https://doi.org/10.1002/JAAL.163

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). *spaCy: Industrial-strength natural language processing in Python*. https://doi.org/10.5281/zenodo.1212303

Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. *Applied Sciences, 4*(1), 9. https://doi.org/10.3390/asi4010009

Huang, Y., & Yu, H. (2022). Research on text generation techniques combining machine learning and deep learning. In *Proceedings of the 3rd Asia-Pacific Conference on Image*

Processing, Electronics and Computers (IPEC '22) (pp. 319–326). Association for Computing Machinery. https://doi.org/10.1145/3544109.3544168

Hunston, S. (2006). *Corpus linguistics*. In K. Brown (Ed.), *Encyclopedia of language & linguistics* (2nd ed.). Elsevier. https://doi.org/10.1016/B0-08-044854-2/00944-5

Ilagan, J. B. R., Alabastro, Z. M. C., Basallo, C. L., & Ilagan, J. R. S. (2024, February). Exploratory customer discovery through simulation using ChatGPT and prompt engineering. In *Proceedings of the 9th International Congress on Information and Communication Technology* (London, United Kingdom). Ateneo Business Insights Laboratory for Development (BUILD). Retrieved from https://www.researchgate.net/publication/379986031_Exploratory_customer_discovery_through_simulation_using_ChatGPT_and_prompt_engineering

Imperial, J. M., & Madabushi, H. T. (2022). *Uniform complexity for text generation*. arXiv. https://arxiv.org/abs/2204.05185

Imperial, J. M., & Madabushi, H. T. (2023). *Flesch or Fumble? Evaluating readability standard alignment of instruction-tuned language models* (arXiv:2309.05454). https://doi.org/10.48550/arXiv.2309.05454

Jana, S., et al. (2024). *The evolution and impact of large language model systems: A comprehensive analysis*. Retrieved from https://www.researchgate.net/publication/379091956

Jang, J., Ye, S., & Seo, M. (2022). *Can large language models truly understand prompts? A case study with negated prompts*. Workshop on Transfer Learning for Natural Language Processing, New Orleans. arXiv:2209.12711.

Jeong, C. (2024). *Fine-tuning and utilization methods of domain-specific LLMs*. *Journal of Intelligence and Information Systems*, 30(1), 93–112. https://doi.org/10.13088/jiis.2024.30.1.093

Jhamtani, H., et al. (2021). *Challenges of figurative language understanding in dialogue systems*. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 3871-3883. https://aclanthology.org/2021.emnlp-main.592.pdf

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Chen, D., Dai, W., Shu, H., Madotto, A., & Fung, P. (2024). *Survey of hallucination in natural language generation*. arXiv. https://arxiv.org/abs/2202.03629

Jiang, M., Liu, K. Z., Zhong, M., Schaeffer, R., Ouyang, S., Han, J., & Koyejo, S. (2024). *Investigating data contamination for pre-training language models*. arXiv. https://doi.org/10.48550/arXiv.2401.06059arXiv

Jurafsky, D., & Martin, J. H. (2024). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models* (3rd ed., draft, January 12, 2025). https://web.stanford.edu/~jurafsky/slp3/

Juzek, T. S., & Ward, Z. B. (2025a). *Word overuse and alignment in large language models: The influence of learning from human feedback*. arXiv. https://arxiv.org/abs/2508.01930

Juzek, T. S., & Ward, Z. B. (2025b). *Why does ChatGPT 'delve' so much? Exploring the sources of lexical overrepresentation in large language models*. arXiv. https://arxiv.org/abs/2412.11385

Kamruzzaman, M., & Kim, G. L. (2024). *Prompting techniques for reducing social bias in LLMs through System 1 and System 2 cognitive processes* (arXiv:2404.17218v1). https://arxiv.org/abs/2404.17218v1

Kandra, F., Demberg, V., & Koller, A. (2025). *LLMs syntactically adapt their language use to their conversational partner*. arXiv. https://arxiv.org/abs/2503.07457

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., & Suchomel, V. (2014). *The Sketch Engine: Ten years on*. Lexicography, 1, 7–36. http://www.sketchengine.eu

Kilgarriff, A., Rychlý, P., Smrž, P., & Tugwell, D. (2004). *The Sketch Engine*. In Proceedings of the 11th EURALEX International Congress (pp. 105–116). http://www.sketchengine.eu

Kirchner, J. H., Ahmad, L., Aaronson, S., & Leike, J. (2023, January 31). *New AI classifier for indicating AI-written text*. OpenAI. https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text

Kleinberg, B., Zegers, J., Festor, J., Vida, S., Präsent, J., Loconte, R., & Peereboom, S. (2024). *Trying to be human: Linguistic traces of stochastic empathy in language models*. arXiv. https://doi.org/10.48550/arXiv.2410.01675

Kmainasi, M. B., Khan, R., Shahroor, A. E., Bendou, B., Hasanain, M., & Alam, F. (2024). *Native vs non-native language prompting: A comparative analysis*. arXiv. https://arxiv.org/abs/2409.07054v1

Kobayashi, G., Kuribayashi, T., Yokoi, S., & Inui, K. (2023). Transformer language models handle word frequency in prediction head. In *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 4523–4535). Association for Computational Linguistics. https://aclanthology.org/2023.findings-acl.276/

Koo, Y., Lee, J., Park, D., Park, S., & Lee, S. (2025, February 16). *Evaluating large language models on understanding Korean indirect speech acts*. arXiv. https://arxiv.org/abs/2502.10995

Kryściński, W., McCann, B., Xiong, C., & Socher, R. (2019). Evaluating the factual consistency of abstractive text summarization. *arXiv*. https://doi.org/10.48550/arXiv.1910.12840

Kyle, K., & Crossley, S. A. (2018). Measuring syntactic complexity in L2 writing using fine-grained clausal and phrasal indices. The Modern Language Journal, 102(2), 333–349. https://doi.org/10.1111/modl.12468

Kyle, K., Crossley, S., & Verspoor, M. (2021). Measuring longitudinal writing development using indices of syntactic complexity and sophistication. Studies in Second Language Acquisition, 43(4), 781–812. https://doi.org/10.1017/S0272263120000546

Lamothe, M., Guéhéneuc, Y.-G., & Shang, W. (2021). A systematic review of API evolution literature. *ACM Computing Surveys, 54*(8), Article 171, 1–36. https://doi.org/10.1145/3470133

LangChain. (2025). *LangSmith: A unified observability & evals platform*. LangChain. Retrieved from https://www.langchain.com/langsmith

Langfuse. (2025). *Langfuse: Open-source LLM engineering platform*. Langfuse. Retrieved from https://langfuse.com/

Leidinger, A., van Rooij, R., & Shutova, E. (2023). *The language of prompting: What linguistic properties make a prompt successful?* arXiv. https://arxiv.org/abs/2311.01967

Leidinger, A., van Rooij, R., & Shutova, E. (2023). *The language of prompting: What linguistic properties make a prompt successful?* In *Findings of the Association for Computational Linguistics: EMNLP 2023* (pp. 9210–9232). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-emnlp.618

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, 2021, 3045–3060. https://doi.org/10.18653/v1/2021.emnlp-main.242

Li, J., Tang, T., Zhao, W. X., Nie, J.-Y., & Wen, J.-R. (2024a). *Pre-trained language models for text generation: A survey*. *ACM Computing Surveys, 56*(9), Article 230, 39 pages. https://doi.org/10.1145/3649449

Li, X., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 2021 Annual Meeting of the Association for Computational Linguistics (ACL 2021)*, 2021, 2757–2772. https://doi.org/10.18653/v1/2021.acl-long.353

Li, Y. (2023, September 22). A practical survey on zero-shot prompt design for in-context learning. *In Proceedings of [Conference/Workshop]*. https://doi.org/10.26615/978-954-452-092-2_069

Li, Y. B., & Wu, K. (2023). SPELL: Semantic prompt evolution based on a large language model. *arXiv*. https://arxiv.org/abs/2310.01260

Li, Z., Ji, J., Ge, Y., Hua, W., & Zhang, Y. (2024b). PAP-REC: Personalised Automatic Prompt for Recommendation Language Model. *arXiv*. https://arxiv.org/abs/2402.00284

Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., & Zou, J. (2023). *GPT detectors are biased against non-native English writers* (Preprint). arXiv. https://doi.org/10.48550/arXiv.2304.02819

Lin, C.-Y. (2004). *ROUGE: A package for automatic evaluation of summaries*. In *Text summarization branches out* (pp. 74–81). Association for Computational Linguistics. https://aclanthology.org/W04-1013/

Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring how models mimic human falsehoods. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3214–3252. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.229

Lin, Z. (2024). Prompt engineering for applied linguistics: Elements, examples, techniques, and strategies. *English Language Teaching, 17*(9), 14. https://doi.org/10.5539/elt.v17n9p14

Linzbach, S., Dimitrov, D., Kallmeyer, L., Evang, K., Jabeen, H., & Dietze, S. (2024). *Dissecting paraphrases: The impact of prompt syntax and supplementary information on knowledge retrieval from pretrained language models*. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Vol. 1, pp. 3645–3655). Association for Computational Linguistics. https://arxiv.org/abs/2404.01992

Lison, P., & Tiedemann, J. (2024). *OpenSubtitles: Multilingual parallel corpora from movie subtitles*. OPUS. Retrieved April 29, 2025, from https://opus.nlpl.eu/OpenSubtitles/corpus/version/OpenSubtitles

Liu, E., Cui, C., Zheng, K., & Neubig, G. (2022). Testing the ability of language models to interpret figurative language. *arXiv*. https://arxiv.org/abs/2204.12632

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021, July 28). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv*. https://doi.org/10.48550/arXiv.2107.13586

Liu, Y., Cheung, A. K. F., & Liu, K. (2023). Syntactic complexity of interpreted, L2, and L1 speech: A constrained language perspective. Lingua, 286, 103509. https://doi.org/10.1016/j.lingua.2023.103509

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., & Zhu, C. (2023). *G-Eval: NLG evaluation using GPT-4 with better human alignment*. arXiv. https://doi.org/10.48550/arXiv.2303.16634 web3.arxiv.org+1arXiv+1

Liu K, Afzaal M. (2021). Syntactic complexity in translated and non-translated texts: A corpus-based study of simplification. PLoS ONE 16(6): e0253454. https://doi.org/10.1371/journal. Pone.0253454

Loper, E., & Bird, S. (2002). *NLTK: The natural language toolkit*. https://doi.org/10.48550/arXiv.cs/0205028

Lu, X. (2011). A corpus-based evaluation of syntactic complexity measures as indices of college-level ESL writers' language development. TESOL Quarterly, 45(1), 36–62. https://doi.org/10.5054/tq.2011.240859

Lüdeling, A., & Kytö, M. (2009). *Corpus linguistics: An international handbook* (Vol. 1). Walter de Gruyter. https://www.researchgate.net/publication/333036626_Corpus_linguistics_An_international_handbook

Maimon, A., & Tsarfaty, R. (2023). *CoheSentia: A novel benchmark of incremental versus holistic assessment of coherence in generated texts*. arXiv. https://doi.org/10.48550/arXiv.2310.16329

Mannapur, S. B. (2025). *Understanding data drift and concept drift in machine learning systems*. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 11*(1), 318–330. https://doi.org/10.32628/CSEIT25111239

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60. https://doi.org/10.3115/v1/P14-5010

Manoharan, M. (2024). API rate limiting mechanisms in SaaS applications: A systematic analysis of DDoS protection strategies. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 10*, 1787–1798. https://doi.org/10.32628/CSEIT241061223

Mason, J. (2023, January). *Mastering AI prompt engineering: Sequencing for pre-trained models*. Zenodo. https://doi.org/10.5281/zenodo.8416070

Maynez, J., Narayan, S., Bohnet, B., & McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1906-1919. https://doi.org/10.18653/v1/2020.acl-main.173

McCarthy, P. M., & Jarvis, S. (2010). MTLD, VOCD, and HD-D: *Behavior Research Methods, 42*(2), 381–392. https://doi.org/10.3758/BRM.42.2.381

McEnery, T., & Hardie, A. (2012). *Corpus Linguistics: Method, Theory and Practice*. Cambridge University Press.

Microsoft. (2025a, April 5). *Evaluation and monitoring metrics for generative AI.* Microsoft Learn. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/evaluation-metrics-built-in?utm_source=chatgpt.com&tabs=warning

Microsoft. (2025b, July 16). *Create a prompt*. Microsoft Learn. https://learn.microsoft.com/en-us/ai-builder/create-a-custom-prompt

Miguelañez, C. (2025, January 7). *Collaborative prompt engineering: Best tools and methods*. Latitude. Retrieved February 13, 2025, from https://latitude-blog.ghost.io/blog/collaborative-prompt-engineering-best-tools-and-methods/

Mirzapour, M., Prost, J.-P., & Retoré, C. (2018). Measuring linguistic complexity: Introducing a new categorial metric. LACompLing 2018 - Symposium on Logic and Algorithms in Computational Linguistics, 95–123. https://doi.org/10.1007/978-3-030-30077-7_5. HAL-02146506.

Mirzapour, M., Prost, J.-P., & Retoré, C. (2018). Measuring linguistic complexity: Introducing a new categorial metric. *LACompLing 2018 - Symposium on Logic and Algorithms in Computational Linguistics*, 95–123. https://doi.org/10.1007/978-3-030-30077-7_5. HAL-02146506.

Montahaei, E., Alihosseini, D., & Soleymani Baghshah, M. (2019). Jointly measuring diversity and quality in text generation models. *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, 90–98. https://doi.org/10.18653/v1/W19-2311

Moretti, F. (2005). *Graphs, maps, trees: Abstract models for a literary history*. Verso. https://books.google.it/books/about/Graphs_Maps_Trees.html?id=YL2kvMIF8hEC

Mu, Y., Wu, B. P., Thorne, W., Robinson, A., Aletras, N., Scarton, C., Bontcheva, K., & Song, X. (2024). *Navigating prompt complexity for zero-shot classification: A study of large language models in computational social science*. In *Proceedings of the 2024 Conference on*

Language Resources and Evaluation (LREC). European Language Resources Association. https://aclanthology.org/2024.lrec-main.1055/

Muktadir, G. M. (2023). *A brief history of prompt: Leveraging language models (through advanced prompting)* (Version 2). arXiv. https://doi.org/10.48550/arXiv.2310.04438

Muñoz-Ortiz, A., Gómez-Rodríguez, C., & Vilares, D. (2024). *Contrasting linguistic patterns in human and LLM-generated news text. Artificial Intelligence Review, 57*(265). https://doi.org/10.1007/s10462-024-10903-2

Murr, L., Grainger, M., & Gao, D. (2023). *Testing LLMs on code generation with varying levels of prompt specificity*. arXiv. https://arxiv.org/abs/2311.07599

Nguyen, X.-P., Aljunied, S. M., Joty, S., & Bing, L. (2024). *Democratizing LLMs for low-resource languages by leveraging their English dominant abilities with linguistically-diverse prompts* (arXiv:2306.11372v2). https://arxiv.org/abs/2306.11372

Nilsson, O., & Yngwe, N. (2022). *API latency and user experience: What aspects impact latency and what are the implications for company performance?* (Bachelor's thesis, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science). KTH Digital Archive. https://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1700288&dswid=-6386

Obeidat, M. M., Haider, A. S., Abu Tair, S., & Sahari, Y. (2024). *Analyzing the Performance of Gemini, ChatGPT, and Google Translate in Rendering English Idioms into Arabic*. ResearchGate. https://www.researchgate.net/publication/374417280_The_Potential_Of_Ai_In_Facilitating_Cross-Cultural_Communication_Through_Translation

Önder, A., Akçapınar, G. (2023). Investigating the effect of prompts on learners' academic help-seeking behaviours on the basis of learning analytics. *Educ Inf Technol 28, 16909–16934*. https://doi.org/10.1007/s10639-023-11872-9

Oni, S. B. (2025). *The Impact of AI on the Translation Industry*. ResearchGate. https://www.researchgate.net/publication/391050035_The_Impact_of_AI_on_the_Translation_Industry

OpenAI. (2023). *GPT-4 technical report*. arXiv. https://doi.org/10.48550/arXiv.2303.08774

OpenAI. (2024). OpenAI o1 system card. OpenAI. Retrieved May 6, 2025, from https://openai.com/index/openai-o1-system-card/

OpenAI. (2025a). *ChatGPT response to a user query about an arithmetic word problem*. OpenAI ChatGPT. Retrieved May 1, 2025, from https://chat.openai.com/

OpenAI. (2025b)**.** *Top tourist attractions in Venice*. OpenAI ChatGPT. Retrieved May 1, 2025, https://chat.openai.com/

OpenAI. (2025c). *Hybrid prompting example: Sport sneakers (Few-shot + Negative Prompting)*. OpenAI ChatGPT. Retrieved May 1, 2025, from https://chat.openai.com/

OpenAI. (2025d). *OpenAI platform overview*. https://platform.openai.com/docs/overview

OpenAI. (2025e). *OpenAI Platform Playground*. OpenAI. Retrieved from https://platform.openai.com/playground

OpenAI. (n.d.). Pricing. OpenAI. Retrieved May 6, 2025, from https://platform.openai.com/docs/pricing

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: A method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311–318. https://doi.org/10.3115/1073083.1073135

Portkey AI. (2024). *Evaluating prompt effectiveness: Key metrics and tools*. Portkey AI. Retrieved February 12, 2025, from https://portkey.ai/blog/evaluating-prompt-effectiveness-key-metrics-and-tools/#:~:text=Evaluating%20prompt%20effectiveness%20means%20assessing,quality%20standards%20for%20production%20use.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., & Webber, B. (2008). *The Penn Discourse TreeBank 2.0*. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)* (pp. 2961–2968). European Language Resources Association (ELRA).

PromptLayer. (2025). *PromptLayer: Your workbench for AI engineering*. PromptLayer. Retrieved from https://www.promptlayer.com/

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). *Stanza: A Python natural language processing toolkit for many human languages*. https://doi.org/10.48550/arXiv.2003.07082

Ramamoorthy, L. (2025). Evaluating generative AI: Challenges, methods, and future directions. *International Journal for Multidisciplinary Research (IJFMR), 7*(1). https://doi.org/10.36948/ijfmr.2025.v07i01.37182

Ramírez-Sánchez, G., Bañón, M., Zaragoza-Bernabeu, J., & Ortiz Rojas, S. (2022). Human evaluation of web-crawled parallel corpora for machine translation. In A. Belz, M. Popović, E. Reiter, & A. Shimorina (Eds.), *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)* (pp. 32–41). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.humeval-1.4aclanthology.org

Rathi, N. (2021). Dependency locality and neural surprisal as predictors of processing difficulty: Evidence from reading times. Proceedings of the Workshop on Cognitive Modelling and Computational Linguistics (pp. 171–176). Online Event, June 10, 2021. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.cmcl-1.21

Rawte, V., Priya, P., Tonmoy, S. M. T. I., Zaman, S. M. M., Sheth, A., & Das, A. (2023). *Exploring the relationship between LLM hallucinations and prompt linguistic nuances: Readability, formality, and concreteness*. arXiv. https://arxiv.org/abs/2309.11064

Rei, R., Faruqui, M., Yogatama, D., Choshen, L., Goldberg, Y., & Sutton, C. (2020). COMET: A neural framework for MT evaluation. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2685–2702. https://doi.org/10.18653/v1/2020.emnlp-main.213

Reinhart, A., Brown, D. W., Markey, B., Laudenbach, M., Pantusen, K., Yurko, R., & Weinberg, G. (2024). Do LLMs write like humans? Variation in grammatical and rhetorical styles. *Proceedings of the National Academy of Sciences, 122*(2025), e2422455122. https://doi.org/10.1073/pnas.2422455122

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering, 3*(1), 57–87. https://doi.org/10.1017/S1351324997001502

Reynolds, L., & McDonell, K. (2021, February 15). Prompt programming for large language models: Beyond the few-shot paradigm. *arXiv*. https://doi.org/10.48550/arXiv.2102.07350

Robillard, M. P., Bodden, E., Kawrykow, D., Mezini, M., & Ratchford, T. (2013). Automated API property inference techniques. *IEEE Transactions on Software Engineering, 39*(5), 613–637. https://doi.org/10.1109/TSE.2012.63

Rosillo-Rodes, P., San Miguel, M. Y Sánchez, D. (2025). Entropy and type-token ratio in gigaword corpora. *Physical Review Research, 7*(3), 033054. https://doi.org/10.1103/PhysRevResearch.7.033054

Sabry, A., AlQudah, A. A., & Shorman, A. (2024). *Leveraging GenAI for lesson plan generation: An interactive mega-prompt approach* [Preprint]. arXiv. https://arxiv.org/abs/2403.12071

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). *A systematic survey of prompt engineering in large language models: Techniques and applications* (arXiv:2402.07927). arXiv. https://arxiv.org/abs/2402.07927

Sarhan, H., Shahrezaye, M., & Hegelich, S. (2025). Navigating representation: Utilizing prompt engineering to minimize representational harms in journalist's image captions. *AI and Ethics*. Advance online publication. https://doi.org/10.1007/s43681-025-00773-x

Sclar, M., Choi, Y., Tsvetkov, Y., & Suhr, A. (2023). *Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting*. arXiv. https://arxiv.org/abs/2310.11324

Sen, S. (2023, November 8). *Evaluate and track your LLM experiments with TruLens*. TruEra. https://truera.com/ai-quality-education/generative-ai-observability/evaluate-and-track-your-llm-experiments-with-trulens/

Serbout, S., El Malki, A., Pautasso, C., & Zdun, U. (2024). API rate limit adoption: A pattern collection. *Proceedings of the 28th European Conference on Pattern Languages of Programs (EuroPLoP '23)*, Article 5, 1–20. Association for Computing Machinery. https://doi.org/10.1145/3628034.3628039

Settaluri, L. S., Doshi, M., Kalyan, T. P., Murthy, R., Bhattacharyya, P., & Dabre, R. (2024). *PUB: A pragmatics understanding benchmark for assessing LLMs' pragmatics capabilities*. arXiv. https://doi.org/10.48550/arXiv.2401.07078arXiv

Shaib, C., Barrow, J., Sun, J., Siu, A. F., Wallace, B. C., & Nenkova, A. (2025). *Standardizing the measurement of text diversity: A tool and a comparative analysis of scores* (Version 2). arXiv. https://arxiv.org/abs/2403.00553v2

Shalevska, E. (2025). *Sentence structure in human and AI-generated texts: A comparative study*. PALIMPSEST/ПАЛИМПСЕСТ, 10(19), 15–24. https://doi.org/10.46763/PALIM25101915sh

Shao, Z., Huang, M., Wen, J., Xu, W., & Zhu, X. (2019). Long and diverse text generation with planning-based hierarchical variational model. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3257–3268. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1321

Shen, Y., Zhang, H., Shen, Y., Wang, L., Shi, C., Du, S., & Tao, Y. (2025). AltGen: AI-driven alt text generation for enhancing EPUB accessibility. In *Proceedings of the 2025 International Conference on Artificial Intelligence and Computational Intelligence* (pp. 78–83). Association for Computing Machinery. https://doi.org/10.1145/3730436.3730449

Singh, A., Gupta, M., Garg, S., Kumar, A., & Agrawal, V. (2024). *Beyond captioning: Task-specific prompting for improved VLM performance in mathematical reasoning*. arXiv. https://arxiv.org/abs/2410.05928

Song, Y., Xu, F., Zhou, S., & Neubig, G. (2024). *Beyond browsing: API-based web agents*. arXiv. https://doi.org/10.48550/arXiv.2410.16464

Straka, M., & Straková, J. (2017). UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 88–99. https://doi.org/10.18653/v1/K17-3009

Tan, Z., Wang, S., Yang, Z., Chen, G., Huang, X., Sun, M., & Liu, Y. (2020). Neural machine translation: A review of methods, resources, and tools. *AI Open, 1*, 5–21. https://doi.org/10.1016/j.aiopen.2020.11.001

Tang, C., Wang, Z., Sun, H., & Wu, Y. (2024). *Large language models might not care what you are saying: Prompt format beats descriptions*. arXiv. https://arxiv.org/abs/2408.08780

Tanprasert, T., & Kauchak, D. (2021). Flesch-Kincaid is not a text simplification evaluation metric. In *Proceedings of the First Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)* (pp. 1–14). Association for Computational Linguistics. https://aclanthology.org/2021.gem-1.1

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: A large-scale dataset for fact extraction and VERification. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 809–819). Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1074

Topol, E. J. (2023). *Prompt engineering as an important emerging skill for medical professionals: Tutorial. Journal of Medical Internet Research*, 25, e50638. https://doi.org/10.2196/50638

Tudino, G., & Qin, Y. (2024). *A corpus-driven comparative analysis of AI in academic discourse: Investigating ChatGPT-generated academic texts in social sciences*. *Lingua, 312*, 103838. https://doi.org/10.1016/j.lingua.2024.103838

Turing. (n.d.). *Fine-tuning large language models*. Turing. Retrieved February 11, 2025, from https://www.turing.com/resources/finetuning-large-language-models

Uchida, S. (2024). Using early LLMs for corpus linguistics: Examining ChatGPT's potential and limitations. *Applied Corpus Linguistics, 4*(1), 100089. https://doi.org/10.1016/j.acorp.2024.100089

UCREL. (n.d.). *CLAWS part-of-speech tagger*. Lancaster University. Retrieved September 9, 2025, from https://ucrel.lancs.ac.uk/claws/

UNESCO. (2021). *Recommendation on the ethics of artificial intelligence.* United Nations Educational, Scientific and Cultural Organization. Retrieved February 17, 2025, from https://www.unesco.org/en/articles/recommendation-ethics-artificial-intelligence

United Nations. (2022). *Principles for the ethical use of AI in the UN system.* United Nations System Chief Executives Board for Coordination. Retrieved February 17, 2025, from https://unsceb.org/sites/default/files/2022-09/Principles%20for%20the%20Ethical%20Use%20of%20AI%20in%20the%20UN%20System_1.pdf

University of Washington, Department of English. (2010, January 8). *Close reading*. https://english.washington.edu/sites/english/files/documents/ewp/ha1.pdf

Vadlapati, P. (2023). *Investigating the impact of linguistic errors of prompts on LLM accuracy*. *ESP Journal of Engineering & Technology Advancements*, 3(2), 150–153. https://doi.org/10.56472/25832646/JETA-V3I6P111

Van Deemter, K., Krahmer, E., & Theune, M. (2005). Real versus template-based natural language generation: A false opposition?. *Computational Linguistics, 31*(1), 15–24. https://doi.org/10.1162/0891201053630291

van der Vlist, F. N., Helmond, A., & Seitz, T. (2022). API governance: The case of Facebook's evolution. *Social Media + Society, 8*(2), 1–14. https://doi.org/10.1177/20563051221086194

van Noord, R., Kuzman, T., Rupnik, P., Ljubešić, N., Esplà-Gomis, M., Ramírez-Sánchez, G., & Toral, A. (2024). *Do language models care about text quality? Evaluating web-crawled corpora across 11 languages*. arXiv. https://doi.org/10.48550/arXiv.2403.08693 arXiv

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. arXiv. https://doi.org/10.48550/arXiv.1706.03762

Vatsal, S., & Dubey, H. (2024). *A survey of prompt engineering methods in large language models for different NLP tasks*. *arXiv*. https://arxiv.org/abs/2407.12994

Wahle, J. P., Ruas, T., Xu, Y., & Gipp, B. (2024). *Paraphrase types elicit prompt engineering capabilities*. arXiv. https://arxiv.org/abs/2406.19898

Wang, A., Cho, K., & Lewis, M. (2020). *Asking and answering questions to evaluate the factual consistency of summaries*. arXiv. https://doi.org/10.48550/arXiv.2004.04228 arXiv+7arXiv+7scholar.google.co.uk+7

Wang, X., Gyawali, B., Bruno, J. V., Molloy, H. R., Evanini, K., & Zechner, K. (2019). Using rhetorical structure theory to assess discourse coherence for non-native spontaneous speech. *Proceedings of the Workshop on Discourse Relation Parsing and*

Treebanking 2019, 153–162. Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-2719

Wang, Z., Chu, Z., Ni, S., & Feng, X. (2024). History, Development, and Principles of Large Language Models—An Introductory Survey. arXiv preprint arXiv:2402.06853. https://arxiv.org/abs/2402.06853

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2023). *Chain-of-thought prompting elicits reasoning in large language models* (arXiv:2201.11903v6). https://arxiv.org/abs/2201.11903v6

Weights & Biases. (2025). *Guides*. https://docs.wandb.ai/guides/

Wells, A., Harper, F., Tremblay, M., & Ogunrinde, V. (2025). Machine learning vs. rule-based NLP in Lisp: A historical and functional comparison. *Journal of Artificial Intelligence Research*, *42*(1), 1–20. https://doi.org/10.1162/0891201053630291

Wu, T., Terry, M., & Cai, C. J. (2022). *AI chains: Transparent and controllable human-AI interaction by chaining large language model prompts* (arXiv:2110.01691v3). arXiv. https://doi.org/10.48550/arXiv.2110.01691

Xu, Z., & Sheng, V. S. (2024). *LecPrompt: A prompt-based approach for logical error correction with CodeBERT. arXiv preprint arXiv:2410.08241*

Xu, Z., Peng, K., Ding, L., Tao, D., & Lu, X. (2024). Take care of your prompt bias! Investigating and mitigating prompt bias in factual knowledge extraction. *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 15552–15565. European Language Resources Association (ELRA) and International Committee on Computational Linguistics (ICCL). https://aclanthology.org/2024.lrec-main.1352

Yaffe, P. (2022). Fog index: Is it really worth the trouble? *Ubiquity, 2022*(October), Article 1, 1–4. https://doi.org/10.1145/3568307

Yin, Z., Wang, H., Horio, K., Kawahara, D., & Sekine, S. (2024). *Should we respect LLMs? A cross-lingual study on the influence of prompt politeness on LLM performance*. arXiv:2402.14531. https://arxiv.org/abs/2402.14531

Yngve, V. H. (1960). A model and hypothesis for language structure. Proceedings of the American Philosophical Association, 104(5), 444–466.

Zappavigna, M. (2023). Hack your corpus analysis: How AI can assist corpus linguists deal with messy social media data. *ACORP*, *1*(1), 100067. https://doi.org/10.1016/j.acorp.2023.100067

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=SkeHuCVFDr

Zhang, Y., Yuan, Y., & Yao, A. C.-C. (2023). *Meta Prompting for AGI Systems*. arXiv. https://arxiv.org/abs/2311.11482

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K.-W. (2018). *Gender bias in coreference resolution: Evaluation and debiasing methods*. arXiv. https://doi.org/10.48550/arXiv.1804.06876

Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C. M., & Eger, S. (2019). MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 563–578. https://doi.org/10.18653/v1/D19-1053

Zhao, W., Strube, M., & Eger, S. (2023). Evaluating text generation with BERT and discourse coherence. Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL), 3865–3883. https://doi.org/10.18653/v1/2023.eacl-main.278

Zheng, J., Qiu, S., & Ma, Q. (2024). *Can LLMs learn new concepts incrementally without forgetting?* arXiv. https://arxiv.org/abs/2402.08526

Zhou, J., Li, Y., & Lin, J. (2024a). Large language model-based question generation aligned with Bloom's Taxonomy [Preprint]. arXiv. https://arxiv.org/abs/2401.05914

Zhou, Y., He, B., & Sun, L. (2024b). Humanizing machine-generated content: Evading AI-text detection through adversarial attack. *arXiv preprint arXiv:2404.01907*. https://doi.org/10.48550/arxiv.2404.01907

Zindela, N. (2023). Comparing measures of syntactic and lexical complexity in artificial intelligence and L2 human-generated argumentative essays. *International Journal of Education and Development using Information and Communication Technology, 19*(3), 50–68. https://www.researchgate.net/publication/377029323_Comparing_Measures_of_Syntactic_and_Lexical_Complexity_in_Artificial_Intelligence_and_L2_Human-Generated_Argumentative_Essays

Zou, L., Carl, M., Mirzapour, M., & Vieira, L. N. (2022). AI-based syntactic complexity metrics and sight interpreting performance. In Lecture Notes in Computer Science (pp. 598–610). Springer. https://doi.org/10.1007/978-3-030-98404-5_49

Zou, L., Carl, M., Mirzapour, M., & Vieira, L. N. (2022). AI-based syntactic complexity metrics and sight interpreting performance. In *Lecture Notes in Computer Science* (pp. 598–610). Springer. https://doi.org/10.1007/978-3-030-98404-5_49

Zwerdling, N., Shlomov, S., Goldbraich, E., Kour, G., Carmeli, B., Tepper, N., Ronen, I., & Zabershinsky, V. (2022). *Understanding the properties of generated corpora*. arXiv. https://arxiv.org/abs/2206.11219

Appendix

List of the prompts used:

Topic	Group 1 Prompts: Simple Prompt	Group 2 Prompts: 2000-Word Prompt	Group 3 Prompts: Advanced (CoT/ToT)
1. Press: Reportage	Invent a film script that focuses on journalism, investigative reporting, or news media.	Invent a film script that focuses on reportage or field journalism. The selected script should involve journalists actively reporting from real-world events, covering social or political issues. Output should be at least 2000 words.	Invent a film script that focuses on journalistic investigations driven by real-world political or social scandals, with strong focus on uncovering hidden truths. Emphasise stories where the process of gathering evidence and the ethical dilemmas of reporting play a central narrative role. The scripts should highlight the tension between media freedom and institutional power. Examples include: Spotlight, All the President's Men. Output should be at least 2000 words. CoT
2. Press: Editorial	Invent a film script that focuses on editorial processes, opinion journalism, or media critique.	Invent a film script that focuses on editorial work and decision-making in media. The script should showcase the behind-the-scenes shaping of news and public opinion.	Invent a film script that focuses on the inner workings of editorial departments, where conflicts arise from political agendas, personal ideologies, or institutional bias. Look for narratives exploring how editorial voices shape public

		Output should be at least 2000 words.	discourse and face backlash or support. The scripts should involve characters navigating complex relationships with truth, representation, and accountability. Examples include: The Post, Good Night, and Good Luck. Output should be at least 2000 words. ToT
3. Press: Reviews	Invent a film script that focuses on critics and film reviews.	Invent a film script that focuses on the central character as a reviewer or critic, ideally of film, literature, or food. Output should be at least 2000 words.	Invent a film script that focuses on critics, their personal journeys, and the social influence of their evaluations. Emphasise narratives where the act of critique affects professional relationships, artistic production, or personal identity. The scripts should reflect on the blurred lines between subjectivity, authority, and cultural relevance. Examples include: Birdman, Ratatouille. Output should be at least 2000 words. CoT
4. Religion	Invent a film script that focuses on religious themes, faith-based narratives, or theological debates.	Invent a film script that focuses on religious beliefs, institutions, or spiritual journeys. Include theological themes or moral dilemmas. Output	Invent a film script that focuses on faith and religious institutions, often through crises of belief, spiritual transformation, or theological debate. Focus on stories that interweave personal convictions with

		should be at least 2000 words.	communal rituals or philosophical tension. The narratives should examine how religious identity influences moral dilemmas and interpersonal dynamics. Examples include: Silence, The Two Popes. Output should be at least 2000 words. ToT
5. Skills, Trades, and Hobbies	Invent a film script that focuses on craftsmanship, specific professions, or hobbies.	Invent a film script that focuses on a unique trade, craft, or hobby, revealing the challenges one can face on this journey. Output should be at least 2000 words.	Invent a film script that focuses on characters deeply engaged in specific crafts or professions, where mastery over a skill reflects broader emotional or narrative arcs. Emphasise the storytelling potential in hands-on environments that require discipline, repetition, and personal expression. These stories should reveal character growth through tangible creation or dedication to a vocation. Examples include: Julie & Julia, Ford v Ferrari. Output should be at least 2000 words. CoT
6. Popular Lore	Invent a film script that focuses on folklore, myths, or legends.	Invent a film script that focuses on folklore, myth, or traditional legend. Output should be at least 2000 words.	Invent a film script that focuses on reinterpreting or reviving folklore, mythology, or cultural legends, often blending them with modern themes or narrative innovation.

			Focus on stories where mythic elements guide the plot or character development, providing allegorical depth. These scripts should highlight cultural transmission and the tension between tradition and reinterpretation. Examples include: Coco, Big Fish. Output should be at least 2000 words. ToT
7. Belles Lettres, Biography, Essays	Invent a film script that focuses on literary works, biographies, or storytelling.	Invent a film script that focuses on adapting from a literary essay, biography, or literary fiction, describing people that really existed, not invented characters. Output should be at least 2000 words.	Invent a film script that focuses on literary lives, philosophical reflection, or essayistic exploration of real events and personalities. Highlight stories that translate internal intellectual landscapes into visual storytelling, often through layered dialogue and memory structures. The narratives should prioritise introspection, artistic ambition, and the impact of ideas. Examples include: A Beautiful Mind, Dead Poets Society. Output should be at least 2000 words. CoT
8. Miscellaneous (Government Documents, Foundation	Invent a film script that focuses on governmental,	Invent a film script that focuses on setting in a government,	Invent a film script that focuses on setting in bureaucratic, industrial, or academic institutions,

Reports, Industry Reports, College, Catalogue, Industry House Organs)	bureaucratic, academic, or corporate environments.	corporate, or academic setting, highlighting challenges people face dealing with institutional systems. Output should be at least 2000 words.	where structural systems and personal ambition collide. Look for narratives that examine the friction between individuality and rules, often through workplace tension or intellectual rivalry. These stories should present the institutional environment as a catalyst for moral or professional crises. Examples include: The Social Network, Erin Brockovich. Output should be at least 2000 words. ToT
9. Learned and Scientific Writings	Invent a film script that focuses on science, academia, or research-based stories.	Invent a film script that focuses on science, academia, or research drives the story, revealing step-by-step development of the character. Output should be at least 2000 words.	Invent a film script that focuses on scientific discovery, academic pursuits, or the emotional consequences of intellectual obsession. Develop a structured narrative that gradually reveals the protagonist's internal and external conflicts as they navigate knowledge-driven environments, such as universities, labs, or isolated research conditions. The story should interweave philosophical reflection, personal sacrifice, and ethical tension, building to a resolution where insight or innovation transforms the individual or society.

			Examples include: A Beautiful Mind, Hidden Figures. Output should be at least 2000 words. CoT
10. General Fiction	Invent a film script that focuses on telling original fictional stories (no genre restrictions).	Invent an original fictional script that doesn't rely on genre conventions. Output should be at least 2000 words.	Invent film scripts that feature original fictional stories outside of fixed genres, focusing on universal human experiences like grief, isolation, memory, or identity. Look for introspective narratives that reveal emotional depth through realistic dialogue and nuanced pacing. These scripts should prioritise psychological realism and emotional resonance over spectacle. Examples include: Nomadland, Marriage Story. Output should be at least 2000 words. CoT
11. Mystery and Detective Fiction	Invent a film script that focuses on crime, detective and mystery.	Invent a film script that focuses on a mystery or detective script with a strong investigative arc. Output should be at least 2000 words.	Invent a film script that focuses on enigmas, missing pieces, or complex crimes, where characters must piece together information under psychological pressure. Emphasise narratives that involve layered storytelling, shifting perspectives, and moral ambiguity. These stories

			should heighten suspense through pacing, red herrings, and unreliable characters. Examples include: Zodiac, Knives Out. Output should be at least 2000 words. ToT
12. Science Fiction	Invent a film script that focuses on futuristic, speculative, or technological themes.	Invent a film script that focuses on futuristic or speculative ideas that can fundamentally change society. Output should be at least 2000 words.	Invent a film script that focuses on future technologies, alternative societies, or philosophical dilemmas rooted in scientific advances. Focus on stories where speculative elements challenge perceptions of identity, ethics, or human potential. These scripts should explore conceptual depth while constructing immersive and internally consistent worlds. Examples include: Interstellar, Arrival. Output should be at least 2000 words. ToT
13. Adventure and Western Fiction	Invent a film script that focuses on classic adventure films or Westerns.	Invent a film script that focuses on adventure or Western genres, where the survival skills of the characters are tested. Output should be at least 2000 words.	Invent a film script that focuses on physical and moral quests through hostile environments, often grounded in survival, frontier justice, or personal redemption. Highlight the use of natural landscapes as emotional terrain, influencing character choices and conflicts.

			These scripts should build momentum through trials that reveal core values or existential reckoning. Examples include: The Revenant, No Country for Old Men. Output should be at least 2000 words. CoT
14. Romance and Love Story	Invent a film script that focuses on romantic relationships and love stories.	Invent a film script that focuses on romantic love, emotional conflict, or gradual relationship development with a conflict. Output should be at least 2000 words.	Invent a film script that focuses on emotionally rich love stories, whether triumphant, doomed, or transformative, in personal or cultural contexts. Focus on the dynamics of intimacy, timing, and vulnerability that shape romantic relationships. These narratives should use structure and dialogue to explore longing, connection, and emotional risk. Examples include: The Notebook, La La Land. Output should be at least 2000 words. ToT
15. Humor	Invent a film script that focuses on comedies and satirical films.	Invent a film script that focuses on humour is the driving force that laughs on societal issues. Output should be at least 2000 words.	Invent a film script that focuses on comedic devices like irony, exaggeration, absurdity, or satire to expose character flaws and social dynamics. Emphasise stories where humour evolves from dialogue, setting, and emotional contrast. The scripts should highlight

			how laughter emerges from discomfort, tension, or surprise. Examples include: Superbad, Dr. Strangelove. Output should be at least 2000 words. CoT

Code for the corpora construction:

```
import os

from openai import OpenAI



client = OpenAI(

    api_key=os.environ.get("OPENAI_API_KEY")

)



# Group 1 prompts

prompts = [

    "Invent a film script that focuses on journalism, investigative reporting, or news media.",

    "Invent a film script that focuses on editorial processes, opinion journalism, or media
critique.",

    "Invent a film script that focuses on critics and film reviews.",

    "Invent a film script that focuses on religious themes, faith-based narratives, or
theological debates.",

    "Invent a film script that focuses on craftsmanship, specific professions, or hobbies.",
    "Invent a film script that focuses on folklore, myths, or legends.",
    "Invent a film script that focuses on literary works, biographies, or storytelling.",
    "Invent a film script that focuses on governmental, bureaucratic, academic, or corporate environments.",
    "Invent a film script that focuses on science, academia, or research-based stories.",
    "Invent a film script that focuses on telling original fictional stories (no genre restrictions).",
    "Invent a film script that focuses on crime, detective and mystery.",
    "Invent a film script that focuses on futuristic, speculative, or technological themes.",
    "Invent a film script that focuses on classic adventure films or Westerns.",
    "Invent a film script that focuses on romantic relationships and love stories.",
    "Invent a film script that focuses on comedies and satirical films.",
]


file = open("group1.txt", "w", encoding="utf-8")


for index, prompt in enumerate(prompts, start=1):
    response = client.responses.create(
        model="o1",
        instructions="You are a helpful assistant",
        input=prompt,
    )
    file.write(f"Prompt {index}:\n")

    file.write(f"Prompt: {prompt}\n\n")

    file.write(response.output_text + "\n\n")



file.close()
```

Code for the metrics:

```
import os

import gc

import spacy

import numpy as np

from collections import Counter



nlp = spacy.load("en_core_web_sm")

nlp.max_length = 5000000



def is_clause_spacy(deprel):

    return deprel in {"acl", "advcl", "ccomp", "xcomp", "csubj",

                      "csubj:pass", "parataxis"}
```

```
# Main function: sentence-by-sentence processing
def process_file_incrementally(filename):
    print(f"\n=== Processing File: {filename} ===")

    idt_values = []
    dlt_values = []
    nnd_values = []
    le_values = []

    # Clause metrics accumulators
    total_sentences = total_clauses = total_words = 0
    total_clause_words = dependent_clauses = coordinate_phrases = complex_noun_phrases
= 0

    with open(filename, "r", encoding="utf8") as infile:
        for line in infile:
            if not line.strip():
                continue  # Skip empty lines

            doc = nlp(line)
            for sent in doc.sents:
                words = list(sent)
```

```
# === DEPENDENCY-BASED METRICS ===


# IDT (Intervening Dependency Theory)
for i in range(len(words) - 1):
    count = 0
    for j, word in enumerate(words):
        head_idx = words.index(word.head) if word.head in words else -1
        if (j <= i and head_idx > i + 1) or (j > i + 1 and head_idx <= i):
            count += 1
    idt_values.append(count)


# DLT (Dependency Locality Theory)
for i, word in enumerate(words):
    if word.pos_ in {"NOUN", "PROPN", "VERB"}:
        left_deps = [j for j, w in enumerate(words) if w.head == word and j < i]
        if left_deps:
            leftmost = min(left_deps)
            dlt = sum(1 for j in range(leftmost, i)
                      if words[j].pos_ in {"NOUN", "PROPN", "VERB"})
            dlt_values.append(dlt)
        else:
            dlt_values.append(0)


# NND (Nearest Noun Dependency)

for i, word in enumerate(words):

    if word.pos_.startswith("NOUN"):

        current = word

        while current.head != current:

            ancestor = current.head

            if ancestor.pos_.startswith("NOUN"):

                distance = abs(words.index(ancestor) - i)

                nnd_values.append(distance)

                break

            if ancestor == ancestor.head:

                break

            current = ancestor


# LE (Left Embedding)

main_verb = next((w for w in words if w.pos_ == "VERB"), None)

if main_verb:

    count = sum(1 for w in words if w.i < main_verb.i and w.pos_ != "VERB")

    le_values.append(count)
```

```
                # === CLAUSE-BASED METRICS ===

                total_sentences += 1

                total_words += len(words)



                clause_count = sum(1 for w in words if w.dep_ == "ROOT" or
is_clause_spacy(w.dep_))

                total_clauses += clause_count

                total_clause_words += len(words)



                dependent_clauses += sum(1 for w in words if is_clause_spacy(w.dep_))

                coordinate_phrases += sum(1 for w in words if w.dep_ == "conj")



                for w in words:

                    if w.pos_ in {"NOUN", "PROPN"}:

                        modifiers = [x for x in words if x.head == w and x.dep_ in {"amod",
"nmod", "acl"}]

                        if modifiers:

                            complex_noun_phrases += 1



    # === Results Summary ===



    print(f"\n--- Dependency-based Metrics for {filename} ---")
```

```
  results_1 = {
      "IDT_SUM": sum(idt_values),
      "IDT_MAX": max(idt_values) if idt_values else 0,
      "IDT_AVG": sum(idt_values)/len(idt_values) if idt_values else 0,
      "DLT_SUM": sum(dlt_values),
      "DLT_MAX": max(dlt_values) if dlt_values else 0,
      "DLT_AVG": sum(dlt_values)/len(dlt_values) if dlt_values else 0,
      "IDT+DLT_SUM": sum(idt_values)+sum(dlt_values),
      "IDT+DLT_MAX": (max(idt_values) if idt_values else 0) + (max(dlt_values) if
dlt_values else 0),
      "IDT+DLT_AVG": (sum(idt_values)+sum(dlt_values)) / len(idt_values + dlt_values)
if (idt_values + dlt_values) else 0,
      "NND_SUM": sum(nnd_values),
      "NND_MAX": max(nnd_values) if nnd_values else 0,
      "NND_AVG": sum(nnd_values)/len(nnd_values) if nnd_values else 0,
      "LE_SUM": sum(le_values),
      "LE_MAX": max(le_values) if le_values else 0,
      "LE_AVG": sum(le_values)/len(le_values) if le_values else 0
  }
  for k, v in results_1.items():
      print(f"{k}: {v:.3f}")
```

```
    print(f"\n--- Clause-based Metrics for {filename} ---")
    results_2 = {
        "MLS": total_words/total_sentences if total_sentences else 0,
        "MLC": total_clause_words/total_clauses if total_clauses else 0,
        "DC/C": dependent_clauses/total_clauses if total_clauses else 0,
        "CP/C": coordinate_phrases/total_clauses if total_clauses else 0,
        "CN/C": complex_noun_phrases/total_clauses if total_clauses else 0,
        "C/S": total_clauses/total_sentences if total_sentences else 0
    }
    for k, v in results_2.items():
        print(f"{k}: {v:.3f}")

    # Memory cleanup
    gc.collect()

# === Main Script: Loop through all .txt files ===
if __name__ == "__main__":
    text_files = [f for f in os.listdir() if f.lower().endswith(".txt")]
    for file in text_files:
        process_file_incrementally(file)
```

Code for the emotion analysis:

```
!pip install transformers


# --- 1. IMPORT LIBRARIES ---
import os
os.environ["WANDB_DISABLED"] = "true"  # Disable Weights & Biases logging


import torch
import pandas as pd
import numpy as np
from transformers import AutoTokenizer, AutoModelForSequenceClassification, Trainer


# --- 2. DEFINE SIMPLE DATASET WRAPPER ---
class SimpleDataset:
    def __init__(self, tokenized_texts):
        self.tokenized_texts = tokenized_texts


    def __len__(self):
        return len(self.tokenized_texts["input_ids"])


    def __getitem__(self, idx):
        return {k: v[idx] for k, v in self.tokenized_texts.items()}
```

```
# --- 3. LOAD MODEL AND TOKENIZER ---

model_name = "j-hartmann/emotion-english-distilroberta-base"

tokenizer = AutoTokenizer.from_pretrained(model_name)

model = AutoModelForSequenceClassification.from_pretrained(model_name)

trainer = Trainer(model=model)


# --- 4. UPLOAD AND READ FILE ---

from google.colab import files

uploaded = files.upload()


file_name = "/content/group3_outputs.txt"  # Make sure file is uploaded with this name


with open(file_name, "r", encoding="utf-8") as f:

    pred_texts = [line.strip() for line in f if line.strip()]


# --- 5. TOKENIZE AND CREATE DATASET ---

tokenized_texts = tokenizer(pred_texts, truncation=True, padding=True,
return_tensors="pt")

pred_dataset = SimpleDataset(tokenized_texts)


# --- 6. MAKE PREDICTIONS ---
```

```
predictions = trainer.predict(pred_dataset)
logits = predictions.predictions
preds = logits.argmax(-1)
labels = pd.Series(preds).map(model.config.id2label)

# --- 7. GET CONFIDENCE SCORES ---
probabilities = np.exp(logits) / np.exp(logits).sum(-1, keepdims=True)
max_scores = probabilities.max(1)

# --- 8. CONSTRUCT DATAFRAME ---
df = pd.DataFrame({
    'text': pred_texts,
    'pred': preds,
    'label': labels,
    'score': max_scores,
    'anger': probabilities[:, 0],
    'disgust': probabilities[:, 1],
    'fear': probabilities[:, 2],
    'joy': probabilities[:, 3],
    'neutral': probabilities[:, 4],
    'sadness': probabilities[:, 5],
    'surprise': probabilities[:, 6],
})


# --- 9. ROUND FOR READABILITY ---
score_cols = ['score', 'anger', 'disgust', 'fear', 'joy', 'neutral', 'sadness', 'surprise']
df[score_cols] = df[score_cols].round(4)


# --- 10. SAVE TO CSV ---
output_file = "/content/group1_SecondTrial_emotions.csv"
df.to_csv(output_file, index=False)


# --- 11. SHOW PREVIEW ---
print("First 10 classified lines:\n")
print(df.head(10).to_string(index=False))


# --- 12. COMPUTE OVERALL EMOTION ---
# Count occurrences
overall_emotion = df['label'].value_counts().idxmax()
print(f"\n Overall dominant emotion in the text: **{overall_emotion.upper()}**")


# --- 13. DOWNLOAD FILE ---
files.download(output_file)
```

Results:

Corpus	MLS	MLC	DC/C	CP/C	CN/C	C/S
Group 1	10.179	7.789	0.243	0.141	0.354	1.307
Group 2	9.496	6.942	0.272	0.114	0.323	1.368
Group 3	9.497	6.977	0.266	0.120	0.383	1.361
Subtitles	7.982	5.811	0.272	0.074	0.123	1.374
Children	21.138	9.799	0.537	0.378	0.337	2.157
Web-UK	22.760	11.265	0.505	0.449	0.648	2.020
Web-COM	21.041	10.428	0.504	0.354	0.537	2.018

Clause-Based Syntactic Complexity Metrics

Corpus	IDT_AVG	DLT_AVG	IDT+DLT_AVG	NND_AVG	LE_AVG
Group 1	5.861	0.223	3.966	3.259	2.963
Group 2	2.707	0.259	1.976	3.309	2.583
Group 3	2.428	0.295	1.791	3.281	2.562
Subtitles	1.499	0.222	1.194	3.255	2.742
Children	2.443	0.342	1.961	4.518	4.286
Web-UK	2.024	0.418	1.576	3.732	4.599

Web-COM	2.025	0.411	1.593	3.755	4.355

Dependency-Based Metrics: Average Values

Corpus	IDT_MAX	DLT_MAX	IDT+DLT_MAX	NND_MAX	LE_MAX
Group 1	66	13	79	21	22
Group 2	66	11	77	49	24
Group 3	62	12	74	34	26
Subtitles	97	177	274	614	389
Children	29	59	88	112	71
Web-UK	32	27	59	52	54
Web-COM	27	25	52	59	59

Dependency-Based Metrics: Maximum Values

Corpus	IDT_SUM	DLT_SUM	IDT+DLT_SUM	NND_SUM	LE_SUM
Group 1	195,679	3,778	199,457	6,804	5,927
Group 2	259,401	10,560	269,961	17,834	18,133
Group 3	224,692	11,620	236,312	19,572	17,672
Subtitles	72,633,255	3,375,872	76,009,127	4,314,832	11,433,032
Children	10,956,151	457,318	11,413,469	855,124	801,612

Web-UK	44,870,382	3,585,806	48,456,188	8,195,647	4,259,970
Web-COM	41,688,546	3,092,516	44,781,062	6,575,705	4,006,591

Dependency-Based Metrics: Sum Values