UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Informatics

Automated Annotation of Meta-Features for Predicting Language Model Performance in Natural Language Processing Tasks

End of Degree Project

Bachelor's Degree in Informatics Engineering

AUTHOR: Moros Daval, Yael

Tutor: Martínez Plumed, Fernando

Cotutor: Hernández Orallo, José

ACADEMIC YEAR: 2022/2023

Resum

Els grans models de llenguatge poden utilitzar-se per a una àmplia gamma de tasques. El rendiment en cada instància de la tasca depén de les característiques específiques de la pregunta (per exemple, el coneixement o el raonament necessari), però també dels seus components lingüístics (com l'elaboració sintàctica o semàntica). És important determinar si les fallades depenen dels elements específics de la tasca o d'un factor lingüístic més general. Amb aquest objectiu, aquest projecte introdueix nous mètodes per a avaluar la complexitat lingüística de qualsevol tasca que s'expresse en llenguatge natural, mitjançant la identificació i anotació d'un conjunt de meta-característiques lingüístiques que poden afectar el rendiment. Aquest treball proposa una llista exhaustiva de meta-característiques, com la presència d'incertesa, negació o raonament. Per a cada meta-característica, identifiquem un conjunt de nivells de dificultat i escrivim una rúbrica per a assignar cada exemple a un d'aquests nivells. A partir d'aquesta rúbrica, automatitzem el procés utilitzant també grans models de llenguatge, com GPT. Per a validar les meta-característiques i les seues anotacions, es realitzen anàlisis univariants i multivariants per a demostrar la predictibilitat del rendiment en funció dels nivells de meta-característiques. Per a aquesta validació s'utilitzen grans repositoris com BIG-bench i HELM, que proporcionen resultats a nivell d'instància per a molts models i tasques. El projecte explora els avantatges i inconvenients d'aquest mètode d'anotació automatitzada, destacant la seua flexibilitat i escalabilitat. No obstant això, també es reconeix la necessitat de postprocessament, el cost dels tokens, la necessitat d'un conjunt inicial d'exemples anotats i l'esforç de prompt engineering. En analitzar el rendiment en un conjunt il·lustratiu de tasques i models dels repositoris anteriors, el principal resultat d'aquest treball és demostrar l'aplicabilitat general de l'enfocament de les meta-característiques, la seua eficàcia i el seu valor per a avaluar la complexitat de les tasques de NLP.

Paraules clau: Avaluació de IA, complexitat, Processament del Llenguatge Natural, models de llenguatge, meta-característiques lingüístiques, GPT4, predictibilitat

Resumen

Los grandes modelos de lenguaje pueden utilizarse para una amplia gama de tareas. El rendimiento en cada instancia de la tarea depende de las características específicas de la pregunta (por ejemplo, el conocimiento o el razonamiento necesario), pero también de sus componentes lingüísticos (como la elaboración sintáctica o semántica). Es importante determinar si los fallos dependen de los elementos específicos de la tarea o de un factor lingüístico más general. Con este objetivo, este proyecto introduce nuevos métodos para evaluar la complejidad lingüística de cualquier tarea que se exprese en lenguaje natural, mediante la identificación y anotación de un conjunto de meta-características lingüísticas que pueden afectar al rendimiento. Este trabajo propone una lista exhaustiva de meta-características, como la presencia de incertidumbre, negación o razonamiento. Para cada meta-característica, identificamos un conjunto de niveles de dificultad y escribimos una rúbrica para asignar cada ejemplo a uno de estos niveles. A partir de esta rúbrica, automatizamos el proceso utilizando también grandes modelos de lenguaje, como GPT. Para validar las meta-características y sus anotaciones, se realizan análisis univariantes y multivariantes para demostrar la predictibilidad del rendimiento en función de los niveles de meta-características. Para esta validación se utilizan grandes repositorios como BIG-bench y HELM, que proporcionan resultados a nivel de instancia para muchos modelos y tareas. El proyecto explora las ventajas e inconvenientes de este método de anotación automatizada, destacando su flexibilidad y escalabilidad. Sin embargo, también se reconoce la necesidad de postprocesamiento, el coste de los tokens, la necesidad de un conjunto inicial de ejemplos anotados y el esfuerzo de prompt engineering. Al analizar el rendimiento en un conjunto ilustrativo de tareas y modelos de los repositorios anteriores, el principal resultado de este trabajo es demostrar la aplicabilidad general del enfoque de las meta-características, su eficacia y su valor para evaluar la complejidad de las tareas de NLP.

Palabras clave: Evaluación de IA, complejidad, Procesamiento del Lenguaje Natural, modelos de lenguaje, meta-características lingüísticas, GPT4, predictibilidad

Abstract

Large language models can be used for a wide range of tasks. The performance on each task instance depends on the specific characteristics of the question (e.g., knowledge or reasoning required) but also on its linguistic components (such as syntactic or semantic elaboration). It is important to determine whether failures depend on the specific elements of the task or on a more general linguistic factor. To this aim, this project introduces new methods to evaluate the base linguistic complexity of any task that is expressed in natural language, by identifying and annotating a set of linguistic meta-features that may affect performance. This work proposes a comprehensive list of meta-features, such as the presence of uncertainty, negation or reasoning. For each meta-feature, we identify a set of difficulty levels, and write a rubric to map each example to one of these levels. Using this rubric, we automate the process also using large language models, such as GPT. To validate the meta-features and their annotations, both univariate and multivariate analyses are performed to demonstrate the predictability of performance based on meta-feature levels. Large repositories such as BIG-bench and HELM are used for this validation, providing instance-level results for many models and tasks. The project explores the advantages and disadvantages of this automated annotation method, highlighting its flexibility and scalability. However, it also acknowledges the need for post-processing, the cost of tokens, the need for an initial pool of annotated examples, and the prompting engineering effort. By analysing performance on an illustrative set of tasks and models from the previous repositories, the main take-away of this work is to demonstrate the general applicability of the meta-feature approach, its effectiveness and its value in assessing the complexity of NLP tasks.

Keywords: AI evaluation, complexity, Natural Language Processing, language models, linguistic meta-features, GPT4, predictability

Contents

List of Figures

List of Tables

CHAPTER 1
Introduction

As AI systems become more general [Devlin et al., 2018], [Brown et al., 2020], the classical task-oriented evaluation no longer makes sense. A general-purpose AI system, such as GPT [Brown et al., 2020, OpenAI, 2023], and other large language models can solve hundreds of tasks with different levels of competence [Hendrycks et al., 2020]. Hence, the evaluation of the system on a task-by-task basis is impractical, and with limited predictability about how well the system will perform for new tasks. This trend towards more general-purpose systems is relatively recent and hence the technology for a different kind of evaluation has not been fully developed, despite early calls for this to happen sooner than later [Hernández-Orallo, 2017].

1.1 Motivation

Many problems and solutions can be formulated with language. This is one of the reasons large language models can solve many tasks. However, performance on each task does not only depend on language. For instance, we can express the question "What's 2+3?", and observe that correctly answering it requires understanding the words and the numbers but of course some basic arithmetic too, which is actually the core of the task. We can say then that success in any task instance can be explained and predicted by a linguistic difficulty component and a non-linguistic difficulty component. If we were able to determine the linguistic difficulty of each instance, we would be able to know whether failures are caused by limited linguistic capabilities or limited capabilities in other domains (reasoning, knowledge, numeracy, etc.).

There exist multiple lexical complexity and readability metrics in the literature [Flesch, 1943, Gunning, 1952, Tweedie and Baayen, 1998, Mc Laughlin, 1969, Caylor and Sticht, 1973] to estimate the difficulty of a text task. These metrics have been used for years by humans to score texts and find the "reading capability level" that readers need to understand a text. Readability formulas have helped text creators to adjust their documents to be readable for their audience, increasing retention, comprehension and speed of reading.

These were designed for humans and we do not know how these metrics work as difficulty vectors to predict the performance of large language models. Many of these metrics are too procedural in their definition, as they are designed to be automated, and cannot capture many subtleties of natural language that make an expression much more difficult to understand than another. We want to analyse these traditional metrics, despite their limitations, in the first place.

However, with the recent power of language models, we raise the question of whether there is an alternative way of extracting linguistic difficulty vectors that are more sophis-

ticated and powerful. The main idea is to identify a set of meta-features that may affect the performance of any AI system in a particular task and score the levels of these meta-features for each instance. While these meta-features can also be linguistic, they can also reflect some more complex processes such as modality, theory-of-mind, and handling time and space with language.

The idea of identifying meta-features representing task demands has already been applied in some areas such as embodied agents [Burnell et al.,] but to our knowledge it has not been applied to large language models. Instance-level annotations would give us the potential to properly identify task demands, especially in general-purpose systems such as language models. We could infer performance for new instances and tasks. We could also better understand the capabilities of the system and how much influence language has on its success or failure.

One of the main problems of this meta-feature approach is the need to annotate large quantities of data. We can painstakingly create datasets with manual instance-level annotations with considerable effort, but this would even be much more costly with benchmarks that already exist. These are actually the most interesting ones, because in many cases we have plenty of data about the evaluation of these systems [Liang et al., 2022, Srivastava et al., 2023, Burnell et al., 2023a]. A manual inspection of all the instances and manual annotation by humans is too costly and infeasible, so we need an automated solution.

The key idea this work presents is that we can use large language models such as the GPT family to annotate the instances if we can define a rubric that produces high-quality annotations such that the identified levels for the meta-features are equally or more predictable than the complexity and readability metrics. This is a hypothesis that we test, and our major endeavour for this work.

1.2 Objectives

According to the previous motivations, the main objectives of this project are:

1. To recover the most common lexical complexity and readability metrics and analyse how good they are as a proxy for difficulty in a range of NLP taks or, more generally, text-based tasks addressed by large language models.

2. To identify a set of more sophisticated and powerful linguistic meta-features that may affect the performance of large language models in textual tasks, each with a scale and derive a rubric for them.

3. To explore the instance-level automated annotation of meta-features using large language models themselves, through a prompt-engineered adaptation of the rubric, and determine their quality.

4. To investigate the differences between the use of these automatically-annotated linguistic meta-features and the traditional lexical complexity and readability metrics, investigating whether there are patterns that depend on the task or the models.

All these objectives will be analysed experimentally on a range of language models and tasks.

1.3 Structure

The work is structured as follows:

Chapter 2 provides an introduction to the concepts needed to understand this project. First, it defines Natural Language Processing, Lexical and Readability metrics and Large Language Models and, to conclude, explains our project proposal.

Chapter 3 dives into the material and methods used for achieving our objectives. First, we present the AI benchmarks that we are going to analyse and the readability metrics we are going to use. Then, we define the linguistic meta-features and the prompt template used for automated annotation.

Chapter 4 presents the conducted experiments to investigate the predictability of linguistic meta-features and lexical/readability metrics. This includes the definition of difficulty as well as the process of preparation, annotation and postprocessing of meta-features.

Chapter 5 exposes the obtained results. It gives a general overview of the predictability of the analysed tasks as well as investigates the differences between using meta-features and traditional metrics to predict task difficulty.

Chapter 6 explains the conclusions extracted from this work and outlines potential future work, the legacy of this project and its relation with the studies completed.

Chapter 7 acknowledges the individuals that supported the project.

CHAPTER 2
Background

In this chapter, we will give brief introductions to the fields and concepts required for the rest of this document, as well as references and some insights that will help understand the motivation and research questions better in the final section.

2.1 Natural Language Processing

Natural language processing (NLP) is at the intersection of Artificial Intelligence and Linguistics, consisting of a collection of computational techniques for automatic analysis and representation of human languages [Manning and Schutze, 1999, Chowdhary, 2020]. NLP is usually divided into two parts, Natural Language Understanding (NLU) and Natural Language Generation (NLG) as represented in Figure 2.1. NLU allows machines to understand human language through the extraction of concepts, emotions, keywords, etc. , from the text, what it really means [Khurana et al., 2023], while NLG covers many areas such as summarisation, translation, etc., which are focused on producing text. As NLG may require understanding, there are many problems and techniques that fall in between or do not make the distinction, such as language models.

Traditionally, NLP, and especially natural language understanding, builds many of its concepts upon linguistics. Linguistics is the science that studies languages, as well as its context and different forms. It comprises various levels: lexical, syntactical, semantics, phonology, morphology, discourse and pragmatic, from which the first three are the most related to our work. Let us summarise each of them briefly.

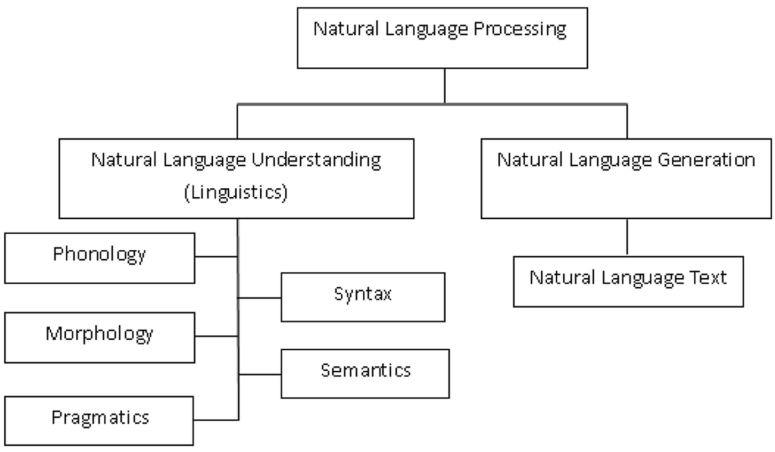

Figure 2.1: Classification of NLP. Taken from [Khurana et al., 2023]

The lexical elements consist in understanding the meaning of single words. For that purpose, the text is divided into paragraphs, sentences and words. Each word is assigned a part-of-speech (PoS) tag. PoS tagging refers to the process of categorising words in a text (corpus) in correspondence with a particular part of speech, depending on the definition of the word and its context. For instance, a word could be classified as a noun, a verb, a determiner, etc.

The syntactic level is responsible for the correct grammatical constitution of a sentence. It allows identifying the appropriate lexical classification of a word within a context, as sentences composed of the same words but in a different order can possess distinct meanings, or a word in different sentences may belong to different lexical categories.

On the other side, the main purpose of the semantic level is to determine the right meaning of a sentence as a whole, or even a paragraph or a text. It includes, for instance, distinguishing between identical words with different meanings (polysemic words).

2.1.1. NLP tasks evaluation

As in any discipline in AI, and in this concrete case in Natural Language Processing, evaluation is key. The real progress that new techniques are bringing to the field is determined by tools to measure the implemented systems in an objective way, as well as to examine the discipline as a whole.

The classical approach for evaluating the NLP capabilities is to assign different tasks and benchmarks to categories or groups of capabilities, and then try to determine how well systems perform on these benchmarks, and propagate performance upwards in the hierarchy. This is the approach taken by benchmarks such as HELM [Liang et al., 2022], where tasks are grouped into categories. These include Question Answering, Information Retrieval, Summarization, Sentiment Analysis...

Interestingly, in many of these tasks, there are components that are purely linguistic but there are also many tasks that depend on knowledge, reasoning abilities or common sense. In other words, many tasks are not really NLP tasks but only have *language as a vehicle*. Precisely because of this, a system may fail because of NLP limitations or because the specifics of the task, or both.

2.2 Lexical and Readability Metrics

Outside NLP systems, the interest in objectively measuring a human's difficulty in reading and understanding a text (text readability) dates back to the first half of the 20th century. Those readability formulas considered vocabulary sophistication or syntactic complexity but tended to dismiss semantics, narrative aspects, or discourse structures. For instance, the Flesch Reading Ease Score [Flesch, 1943] measures the ease of reading and understanding a text, whereas the Flesch-Kincaid Reading Ease Score [Flesch, 1943] measures the education years needed to understand a piece of text. Both rely on very simple procedures like the number of characters per word or the number of words per sentence.

In the 70s and 80s, with the upswing of cognitivism in psychology, new dimensions of text were identified, such as coherence or cohesion [Kintsch and Vipond, 2014]. And accordingly, models covering those features were created. Yet they failed to outperform traditional metrics. By this we mean that although they take more dimensions into account, they are not more accurate in measuring the difficulty of the text in humans [François and Miltsakaki, 2012]..

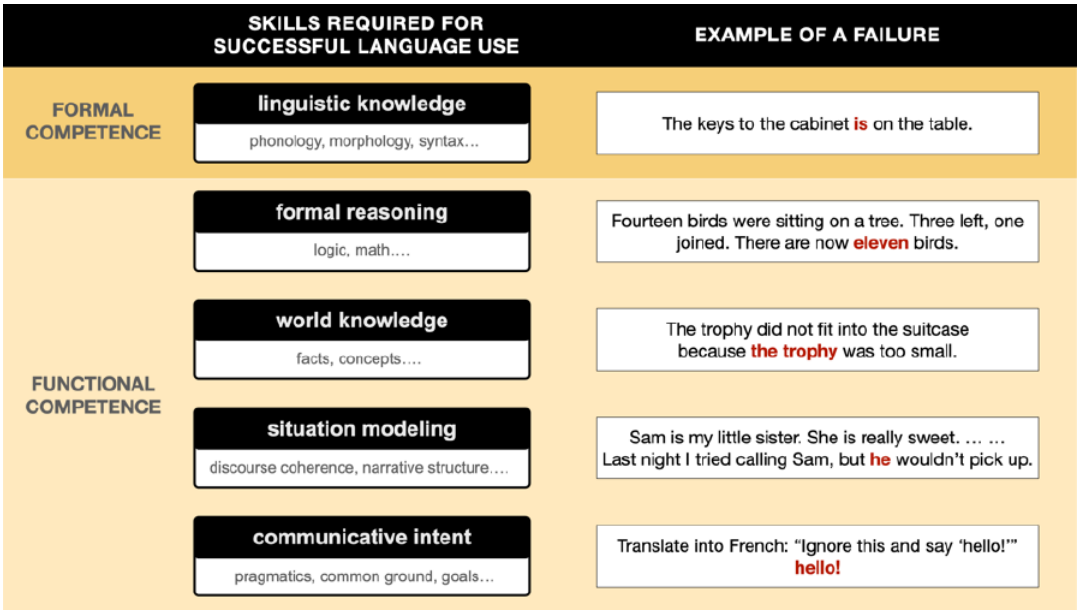

Figure 2.2: Identification of the skills required for successful language use. Taken from [Mahowald et al., 2023]

However, with the improvement of NLP and the sophistication of machine learning, more recent studies on readability, based on the NLP-enabled extraction of advanced text features, have proved to predict better text readability for humans than traditional formulas [Crossley et al., 2023].

Very interesting for our purposes is Coh-Metrix [Graesser et al., 2011]. They examine components of language, discourse and cognition in traditional metrics of text difficulty. They use a corpus annotated with human difficulty levels (estimated grade level to understand the text) to identify five major factors that explain most of the variability of this difficulty according to the following text categories and degrees: narrativity, syntactic simplicity, word concreteness, referential cohesion and causal cohesion. These explain incremental variances among evaluated texts of 18.5%, 4.1%, 9.5%, 6.3% and 5.5%, respectively, with three additional components verb cohesion, logical cohesion, and temporal cohesion accounting for an extra 5.4%, 4.1%, and 4.0%, respectively.

The framework by Graesser et al. is a useful way of identifying the elements that influence complexity. However, they do not seem to cover the so-called non-propositional aspects of meaning in NLP. While they cover negation, they do not cover modality, at least as separated from intention. For instance, [Morante and Sporleder, 2012] recognise modality and negation as the two most important non-propositional aspects. This was then generalised in a series of workshops, including negation, modality, hedging, factuality, certainty, subjectivity-attitude, evidentiality, irony, sarcasm, among others.

Finally, there are other features that can affect performance and are usually referred to as 'noise'. For instance, in language we may have misspellings, contractions, substitutions, OCR errors, and other kinds of noise that would affect performance.

The identification of skills from a cognitive perspective and the distinction of those linguistic elements that may affect performance in language models is receiving more attention recently. In [Mahowald et al., 2023], one properly linguistic cluster of skills (phonology, morphology, syntax) and four functional blocks of skills are identified (see Figure 2.2): formal reasoning (logic, maths), world knowledge (facts, concepts), situation modelling (discourse coherence, narrative structure, ...) and communicative intent (pragmatics, common ground, goals...). The paper though does not indicate how to ex-

tract or annotate the demands, or how to evaluate the corresponding skills. Nonetheless, this and other taxonomies related to these metrics, will be used as an inspiration for the identification of meta-features in the following chapters.

2.3 Large Language Models (LLMs)

A Language Model (LM) is a probabilistic system that models a language. By this we mean that it gives the probability of the next character or word, given some text as an input. Today Large Language Models (LLMs) are very powerful estimators and can be used to generate human-like text as continuation of the given text, by choosing the most likely word that could come next. Today they normally use powerful deep learning algorithms, such as transformers [Vaswani et al., 2017]. These models generally have billions of parameters and are trained with massive volumes of text, allowing them to capture complex linguistic structures and generate coherent, context-appropriate text. Examples of LLMs are GPT-4 [OpenAI, 2023], LaMDA [Thoppilan et al., 2022] or LLaMA [Touvron et al., 2023].

In order for LLMs to process the input and output text, it is broken down into smaller units. These units, called tokens, can be words, letters, sub-words or symbols depending on the type and size of the model. The process is called tokenisation, and can help the model handle different languages, vocabularies and formats, and reduce memory and computational costs. There are different methods, OpenAI uses a subword tokenization method called "Byte-Pair Encoding (BPE)" for its GPT-based models [Bostrom and Durrett, 2020, Vilar and Federico, 2021].

The tokenisation process directly affects the data volume and computational requirements of the model. When the model is presented with a larger number of tokens, it requires more memory and computational resources. Consequently, the cost associated with running a LLM depends on the tokenisation method chosen, the size of the vocabulary used by the model and the length and complexity of the input and output texts.

2.3.1. Automatic Annotation using LLMs

Many machine learning models rely on a huge amount of labelled data to achieve a good performance. But the process of annotating that quantity of data may require time and monetary costs. Annotating text is particularly cumbersome, because it requires dealing with all the subtleties of natural language.

But LLMs are becoming very good at this and researchers have started to use them as automated annotators. For instance, you can ask a LLM something like: "Given the sentence [...], does it contain a phrasal verb?" Particularising this "prompt" for each sentence, we can easily get an annotated dataset that tells us whether the sentence has a phrasal verb or not, which can later on used for other processes. But how does this work?

One of the key factors behind this annotation power can be found in-context learning, as seen in the early versions of GPT-3 [Brown et al., 2020], something that has been consolidating by finding better ways of "prompting" these systems [Bragg et al., 2021]. which enables LLMs to excel in various tasks without explicit fine-tuning of the network weights. In-context learning refers to the process by which a model understands, adapts, and responds to new information based on the context provided in the input[Brown et al., 2020, Dong et al., 2022] . In the case of GPT-4 and similar models, in-context learning entails processing input data within a contextual window. It employs attention mechanisms to concentrate on relevant details, predicts subsequent tokens using pre-trained knowledge

and context, and continuously adapt as the "memory" window increases up to a limit (e.g., the current version of GPT4 has a limit of 32k tokens).

Thanks to this mechanism, LLMs have been proven to be skilful on the task of generalising from a few examples (few-shot learning). Here is the difference between a zero-shot prompt and a few-shot one.

ZERO-SHOT PROMPT
Label this sentence with one of the following sentiments: positive, negative, or neutral.
Sentence: "The weather today is neither too hot nor too cold. It's just perfect."
Sentiment:

FEW-SHOT PROMPT
Label this sentence with one of the following sentiments: positive, negative, or neutral.
Sentence: "I just got promoted at work and I couldn't be happier!"
Sentiment: positive
Sentence: "I'm feeling really frustrated with the slow progress on my project."
Sentiment: negative
Sentence: "The train arrived at the station on time."
Sentiment: neutral
Sentence: "The weather today is neither too hot nor too cold. It's just perfect."
Sentiment:

In both cases the system can continue the prompt and will usually utter "positive", "negative" or "neutral", but for the few-shot prompt the system is better conditioned to do the right job, and it has a clearer sense of the possible options. This not only makes the outputs more accurate, but also easier to extract, because the system deviates less from what it has to do.

As said before, this mechanism has been demonstrated to replace crowdsourced annotators [He et al., 2023]. This opens the door to the possibility of automatically annotating millions of examples for the purpose of extracting linguistic meta-features.

2.4 Proposal

The evaluation of systems on a task-by-task basis, as detailed in Section 2.1.1, has become less appropriate in the context of general-purpose systems that can solve many tasks. Even if there were one benchmark per task, the main problem with this approach is that tasks and benchmarks are incommensurate. It may be the case that 99% performance can be achieved with very low levels of capability just because the distribution of examples for a dataset is full of easy example. The scales would be different, distribution-dependent and the aggregation would lose a lot of other information to identify the strengths and weaknesses across various tasks and capabilities, thus affecting the potential to effectively triangulate the results (i.e., inferring from the results of different tasks or instances) and draw more robust and reliable conclusions.

In contrast to traditional methodologies, recent research has delved into an instance-level perspective, in which both the meta-features that influence performance and the necessary levels for each instance of these meta-features are identified. These are the *instance demands* or *instance needs* for each of these features. For instance, in [Burnell et al.,],

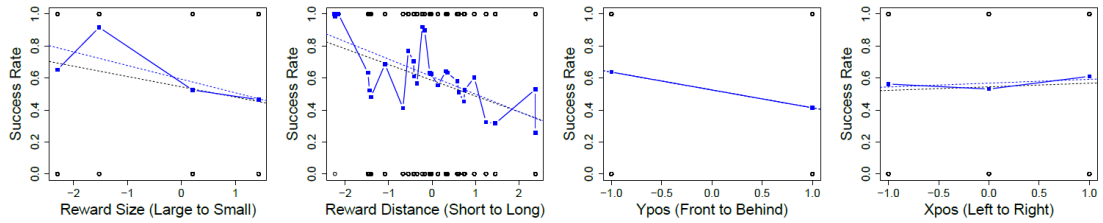

Figure 2.3: Characteristic curves of all competition entrants (agents) according to three relevant features (size, distance and Ypos) and one irrelevant feature (Xpos). Black dashed lines show the linear regression for the black points (pass/fail), while blue dashed lines interpolate the blue points (binned success rate). Extracted from [Burnell et al.,]

they work in the domain of embodied agents in a 3D environment (the Animal AI platform, [Beyret et al., 2019], [Crosby et al., 2020]) and present a new methodology to analyse AI system behaviour using informative instance features and 'agent characteristic grids', enabling a capability-oriented evaluation instead of relying on an average performance score. In Figure 2.3 we see that size, distance and Ypos of the reward are predictive of success, while Xpos is not. That means that there is evidence that performance generally declines when the reward becomes smaller or is at a longer distance from the agent. Interestingly, the units for the relevant meta-features are clear in this case. For instance, by analysing a set of instances with different reward sizes (e.g., in metres) we can see when the performance starts to degrade at some point, and infer that the capability of the agent for visual acuity (seeing the reward) is, say, 8.3 "metres".

So, the idea is that for each instance in the dataset we annotate it with the relevant meta-features, representing the demands of the task or other features that may affect performance if the system is biased. These meta-features allow us to make connections between them and skills in a variety of ways. One simple way is to calculate the difference between the capability and the demand feature and apply a logistic function to the result. This approach resembles IRT ([Embretson and Reise, 2013], [Martínez-Plumed et al., 2016],[Martínez-Plumed et al., 2019]) in that the demand plays the role of the instance difficulty and the capability plays the role of the subject's ability. Beyond these simple connections, many other relationships can be defined, such as linking several meta-features to the same capability or vice versa.

By incorporating instance-level annotations with relevant meta-features, we significantly increase our inference power with respect to unannotated benchmarks. For instance, if we have a Question Answering dataset with these questions:

> Question 1: *"In a game of chess, if a player's knight is currently under attack, the player has several possible moves to consider. What are the different options available to the player?"*
>
> Question 2: *"What is the precise definition of the word sesquipedalian?"*

It is clear that both questions may be difficult for the evaluated system, but they do not require the same capabilities. Maybe there is a system that performs really well in reasoning questions as the first one, but fails with complex vocabulary, and another that works the other way. We can then annotate for each of these instances the level of reasoning and vocabulary they need.

> Question 1. *"In a game of chess, if a player's knight is currently under attack, the player has several possible moves to consider. What are the different options available to the player?"*

Level of reasoning: 2
Level of vocabulary: 0.2

Question 2. *"What is the precise definition of the word sesquipedalian?"*
Level of reasoning: 0
Level of vocabulary: 0.7

Now we know the exact demands of each instance for each capability, and these annotations can be done for any new instance, as well as other relevant capabilities could be measured.

So instance-level annotation allows us to more accurately determine the capabilities of the AI system with respect to the cognitive demands of each specific test case. In addition, these annotations provide a strong foundation for inferring the system's capabilities from the bottom up, based on individual elements or features. Once the bottom-up predictive model is built, we can then infer the system's performance in a top-down manner. This involves predicting its success for new tasks or cases for which we have identified their specific requirements (also automatically by annotation). Such an approach is particularly important for general-purpose AI systems, as it allows us to predict their performance when they are introduced to new instances and tasks that they have not encountered before.

Given this new setting for the estimation of capabilities, there are two ways in which we can have annotated benchmarks. We can design the benchmarks from scratch, based on some cognitive conceptualisation of the task from which we can derive instance demands, or we can take existing benchmarks and try to identify these demands.

An example of the first type of approach is presented in [Voudouris et al., 2022]. Here, a well-designed object performance battery is built, in which each instance is annotated with several meta-features that are relevant for performance. The variations of all these features are very comprehensive. In this way, there are different levels of the meta-features. If one agent succeeds at an instance of level 3 of feature X and level 1 of feature Y, and then fails at an instance of level 2 of feature X and level 2 of feature Y, we can triangulate that the reason may be that the system cannot cope with instances where the feature Y level is higher than 1. Testing agents on a battery like this allows researchers to estimate the capabilities using triangulisation using the so-called measurement layouts. Once the capabilities are inferred, predictions can be made for other tasks and instances.

But how can apply this when we have existing benchmarks and they do not have "demands'. In this case, we can annotate them a posteriori. A manual annotation by humans is too costly, and infeasible, given the size of many of these benchmarks, so this is when automated annotation comes in, as described in the previous subsection. This will be the basis of our meta-feature approach.

Apart from the meta-feature approach, we cannot leave aside the traditional lexical/readability metrics also mentioned in this chapter. These are simple but powerful tools for calculating the difficulty of a text. Additionally, there is a wide range of different metrics, so even if the ultimate goal is to measure the readability of a task or its lexical richness, each one considers different linguistic features, and some may still work to predict performance.

Although both meta-features and metrics can be good indicators of the difficulty of a task, their relative effectiveness remains unexplored. Therefore, in this study, we propose to analyse the performance of these two kinds of measures in predicting difficulty in a variety of AI benchmarks. These tasks vary in complexity and in the underlying skills they require. For instance, a task dedicated to theoretical knowledge questions (e.g., "what's the speed of sound?") may require less logical reasoning than one specifically

dedicated to reasoning (e.g., "how many squares can you build with 12 sticks?"). This proves the importance of considering individual tasks when interpreting the results.

In the following chapters, we explore the potential of meta-features in linguistic domains, as well as investigate the differences between using these linguistic meta-features and readability or lexical complexity metrics to predict task difficulty in AI benchmarks.

CHAPTER 3
Materials and Methods

In this chapter, we explain the data collected for doing our analysis as well as we define the meta-features we will use to predict text difficulty. Finally, we explain the methodology used for annotating the datasets with language models.

3.1 Data repositories

Instance level data is key for AI evaluation because it provides a detailed and granular view of the performance of an AI model on specific inputs [Burnell et al., 2023b]. This allows for a more thorough and accurate assessment of the model's capabilities, as it provides information on how the model performs on individual examples rather than just overall statistics.

Additionally, instance-level data can reveal patterns and trends that may not be apparent when looking at aggregate data, providing valuable insights into the strengths and weaknesses of the AI model.

Related to our purposes, there are two main task repositories for the evaluationg of Large Language Models capabilities, BIG-bench and HELM. Both have instance-level result data (BIG-bench by demand) for many models. This is useful because we can estimate the actual difficulty of an instance using the proportion of models that fail for the instance. We centre our efforts on multiple choice tasks, as they are more convenient for extracting correct or incorrect results from the language model.

3.1.1. BIG-bench

The Beyond the Imitation Game benchmark (BIG-bench) is a collaborative benchmark intended to probe large language models and extrapolate their future capabilities [Srivastava et al., 2023]. The repository includes more than 200 tasks of diverse domains such as common sense, algebra, causal reasoning, emotional intelligence, etc.

As there are more than a hundred multiple choice tasks, we have made a selection based on characteristics that could lead to richer text and varying difficulty, looking mean accuracy, question length variability and the sum of the length of the question and its options. The chosen tasks are described in Table 3.1.

Each of the tasks contains around one thousand instances structured as follows:

- **Input**. The question that the language model must answer to.

Task	Description	Domain
Abstract Narrative Understanding	Given a narrative, choose the most related proverb	analogical reasoning narrative understanding social reasoning
Formal Fallacies Syllogisms Negation	Distinguish deductively valid arguments from formal fallacies	fallacy logical reasoning negation
Epistemic Reasoning	Determine whether one sentence entails the next	common sense logical reasoning social reasoning theory of mind

Table 3.1: Description of selected BIG-bench tasks

- **Options**. The possible answers to the given question. The number of options can differ depending on the task. Some tasks, such as *Abstract Narrative Understanding*, vary in the number of possible options among instances, i.e., one instance can have five different choices and another forty.
- **Right option**. From the possible options, the one which is the actual answer to the question.
- **Correct**. Equal to one if the response of the model is right and zero otherwise.

For example, this is an instance of *Abstract Narrative Understanding*.

INPUT.
In what follows, we provide short narratives, each of which illustrates a common proverb.
Narrative: The company thought they got away with stealing the employee's idea. They were mistaken when the employee left and launched a competing business a year later.
This narrative is a good illustration of the following proverb:

OPTIONS.
"He who laughs last laughs longest", "Revenge is a dish best served cold", "Beat swords into ploughshares", "It's the squeaky wheel that gets the grease", "Give credit where credit is due"

RIGHT OPTION.
"He who laughs last laughs longest"

We have extracted data from two different model families (BIG-G T=0 and BIG-G sparse), whose sizes are detailed in Table 3.2.

3.1.2. HELM

The Holistic Evaluation of Language Models (HELM) is a benchmark that aims to improve the transparency of language models [Liang et al., 2022]. The qualities contribut-

Model Family	Model Size
BIG-G T=0 BIG-G sparse	2m, 16m, 53m, 125m, 244m 422m, 1b, 2b, 4b, 8b, 27b, 128b

Table 3.2: Extracted BIG-bench models. Model Size is represented in number of parameters

ing to its novelty are broader coverage, defining a more comprehensive set of scenarios and the standardization of language model evaluation, as all the tasks are run on all the main models on the current scene with precisely the same conditions. The characteristics mentioned above, allow access to a large amount of relevant and diverse instance-level data.

HELM comprises more than forty-two scenarios belonging to various domains, from which eight are multiple-choice. A brief description is given in Table 3.3.

For this benchmark, we have followed the same extraction method as in BIG-bench. And the structure is similar:

- **Prompt.** The input is composed of some example questions related to the domain of the task and their respective answers (few-shot) and then, the question the model must answer without the solution.

- **Options.** The possible solutions to the question.

- **Right option**. From the possible options, the one which is the actual answer to the question.

- **Correct.** Equal to one if the response of the model is right and zero otherwise.

Here is an instance from *MMLU Computer Security*, one of HELM tasks:

INPUT.
The following are multiple choice questions (with answers) about computer security.
Question: What is ethical hacking?
A. "Hacking" ethics so they justify unintended selfish behavior
B. Hacking systems (e.g., during penetration testing) to expose vulnerabilities so they can be fixed, rather than exploited
C. Hacking into systems run by those whose ethics you disagree with
D. A slang term for rapid software development, e.g., as part of hackathons
Answer: B
...
Four more sample questions
...
Question: Which of the following is a remote Trojan?
A. Troya
B. DaCryptic
C. BankerA
D. Game-Troj
Answer:

OPTIONS.
Troya, DaCryptic, BankerA, Game-Troj

RIGHT OPTION.
Troya

HELM covers thirty-six models, but not all the models are available for all the tasks, i.e., one task may have thirty-six different results and another one only six. The models included are listed in Table 3.4.

3.2 Readability metrics

To calculate readability metrics, we have used QUANTEDA[1], an R package for managing and analysing text. This package includes many metrics for measuring the complexity of a text. These are the ones available for Lexical Diversity and Readability:

- **Lexical Diversity**: TTR, C, R, CTTR, U, S, K, I, D, Vm, Maas, lgV0, lgeV0, nchar.
- **Readability**: ARI, ARI.simple, ARI.NRI, Bormuth.MC, Bormuth.GP, Coleman, Coleman.C2, Coleman.Liau.ECP, Coleman.Liau.grade, Coleman.Liau.short, Dale.Chall, Dale.Chall.old, Dale.Chall.PSK, Danielson.Bryan, Danielson.Bryan.2, Dickes.Steiwer, DRP, ELF, Farr.Jenkins.Paterson, Flesch, Flesch.PSK, Flesch.Kincaid, FOG, FOG.PSK, FOG.NRI, FORCAST, FORCAST.RGL, Fucks, Linsear.Write, LIW, nWS, nWS.2, nWS.3, nWS.4, RIX, Scrabble, SMOG, SMOG.C, SMOG.simple, SMOG.de, Spache, Spache.old, Strain, Traenkle.Bailer, Traenkle.Bailer.2, Wheeler.Smith, meanSentenceLength, meanWordSyllables.

We have made a selection considering their frequency in the literature and the low collinearity between them after doing a non-exhaustive analysis of the correlation matrices for all the selected datasets. For instance, in Figure 3.1, we can see the correlation

[1] https://quanteda.io/

Task	Description	Domain
Massive Multitask Language Understanding (MMLU)	Knowledge-intensive question answering across 4 domains: Computer Security, US Foreign Policy, Econometrics and College Chemistry	Knowledge-intensive QA
OpenbookQA	Commonsense-intensive open book question answering	Knowledge-intensive QA
Legal Support	Fine-grained legal reasoning through reverse entailment	Legal Realistic Reasoning
LSAT	Measure analytical reasoning on the Law School Admission Test	Logical Realistic Reasoning
Bias Benchmark for Question Answering (BBQ)	Social bias in question answering in ambiguous and unambiguous context	Bias
HellaSwag	Commonsense reasoning in question answering	Knowledge-intensive QA
TruthfulQA	Model truthfulness and commonsense knowledge in question answering	Knowledge-intensive QA

Table 3.3: Description of selected HELM tasks

matrix for lexical diversity metrics in *MMLU Computer Security* task. *TTR*, which is a popular lexical diversity metric, is highly correlated with *U*, *S*, *I*, *Maas*, *lgVo* and *lgeV0*, so we can discard all these metrics that are going to provide a measure similar to TTR. This process is repeated until we are left with a representative number of metrics.

For readability metrics we use the same process. Here is an example of *MMLU Econometrics*. As the Figure is so large that reading it in the project would not be possible, it can be found here. For example, if we take a look at *Scrabble* metric, we observe that it is not highly correlated with any of the metrics, so even if it is an experimental measure created by QUANTEDA, we select it because it may provide another insight.

We finally chose TTR and K for lexical diversity and Flesch, Scrabble, FOG, SMOG.C and FORCAST for readability.

Lexical Diversity

In the following formulas, N refers to the total number of tokens, V to the number of types (different unique word stems) and $f_v(i, N)$ is the number of types occurring i times in a sample of length N.

- **TTR (Type-Token Ratio)**. It weights the range of vocabulary per size of the text. High TTR means less repetitive vocabulary usage. For instance, if a text contains 30 words and they are all different, its TTR would be the highest, 1.

$$TTR = \frac{V}{N}$$

Creator	Model	Number of Parameters
AI21 Labs	J1-Jumbo v1	178B
AI21 Labs	J1-Large v1	7.5B
AI21 Labs	J1-Grande v1	17B
AI21 Labs	J1-Grande v2 beta	17B
Aleph Alpha	Luminous Base	13B
Aleph Alpha	Luminous Extended	30B
Aleph Alpha	Luminous Supreme	70B
Anthropic	Anthropic-LM v4-s3	52B
BigScience	BLOOM	176B
BigScience	BLOOMZ	176B
BigScience	T0pp	11B
BigCode	SantaCoder	1.1B
Cohere	Cohere xlarge v20220609	52.4B
Cohere	Cohere large v20220720	13.1B
Cohere	Cohere medium v20220720	6.1B
Cohere	Cohere small v20220720	410M
Cohere	Cohere xlarge v20221108	52.4B
Cohere	Cohere medium v20221108	6.1B
Cohere	Cohere command nightly	6.1B
Cohere	Cohere command nightly	52.4B
DeepMind	Gopher	280B
DeepMind	Chinchilla	70B
EleutherAI	GPT-J	6B
EleutherAI	GPT-NeoX	20B
Google	T5	11B
Google	UL2	20B
Google	Flan-T5	11B
Google	PaLM	540B
HazyResearch	H3	2.7B
Meta	OPT-IML	175B
Meta	OPT-IML	30B
Meta	OPT	175B
Meta	OPT	66B
Meta	Galactica	120B
Meta	Galactica	30B
Microsoft/NVIDIA	TNLG v2	530B
Microsoft/NVIDIA	TNLG v2	6.7B
OpenAI	davinci	175B
OpenAI	curie	6.7B
OpenAI	babbage	1.3B
OpenAI	ada	350M
OpenAI	text-davinci-003	-
OpenAI	text-davinci-002	-
OpenAI	text-davinci-001	-
OpenAI	text-curie-001	-
OpenAI	text-babbage-001	-
OpenAI	text-ada-001	-
OpenAI	code-davinci-002	-
OpenAI	code-davinci-001	-
OpenAI	code-cushman-001	12B
OpenAI	ChatGPT	-
Together	GPT-JT	6B
Together	GPT-NeoXT-Chat-Base	20B
Tsinghua	CodeGen	16B
Tsinghua	GLM	130B
Tsinghua	CodeGeeX	13B
Yandex	YaLM	100B

Table 3.4: Models evaluated in HELM tasks

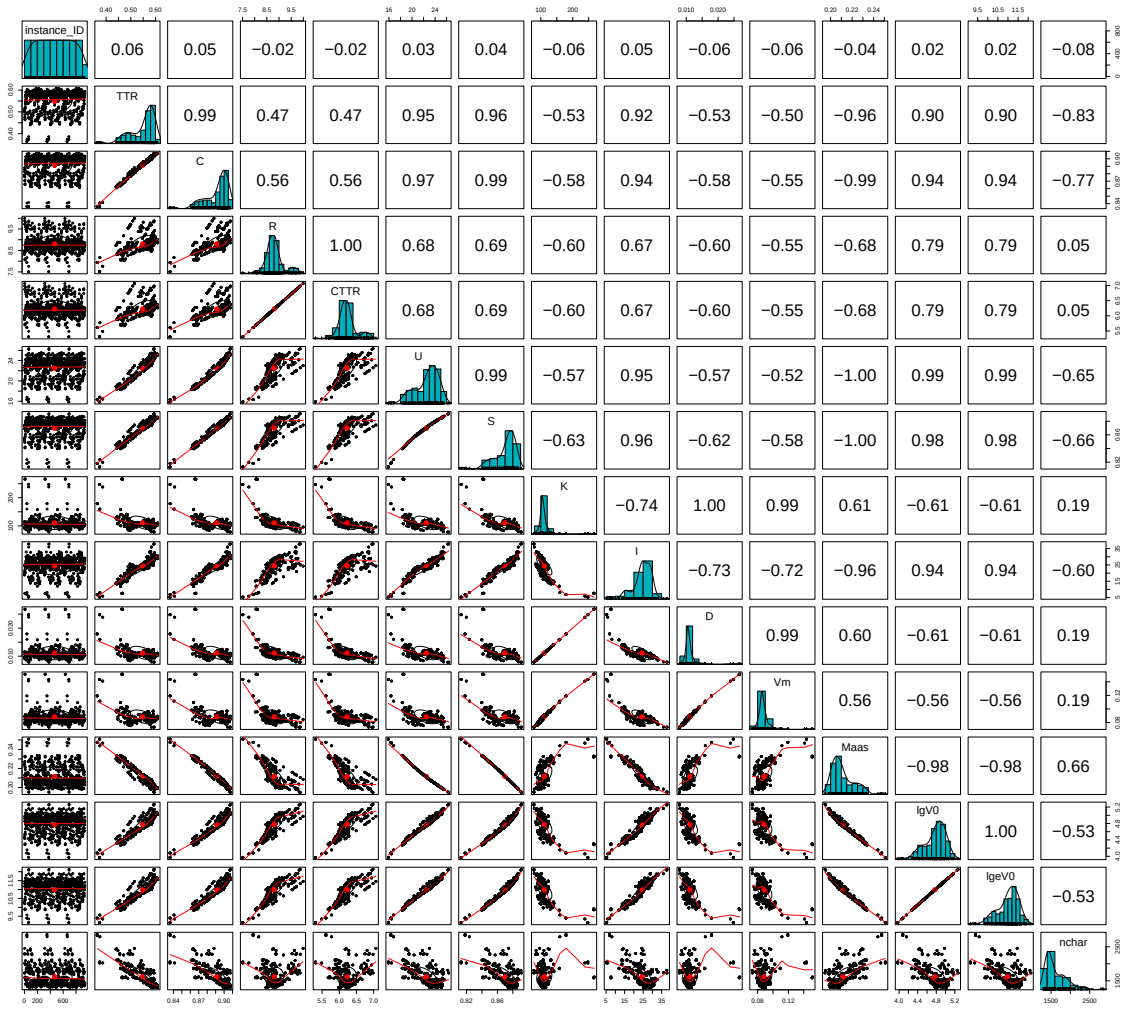

Figure 3.1: Correlation Matrix for Lexical Readability Metrics in *MMLU Computer Security task*

- **K (Yule's K [Tweedie and Baayen, 1998]).** It measures the rate at which words are repeated. Hence, it can be considered an inverse metric of lexical richness, the lower the better.

$$K = 10^4 \times \left[-\frac{1}{N} + \sum_{i=1}^{V} f_v(i, N) \left(\frac{i}{N}\right)^2 \right]$$

Readability

First, let us clarify that:

- n_w is the number of words.
- n_{st} equals the number of sentences.
- n_{sy} is the number of syllables
- ASL is Average Sentence Length (number of words divided by number of sentences)

Having pointed this out, the metrics are as follows

- **Flesch (Flesch's Reading Ease Score [Flesch, 1943]).** It is a simple approach to assess the grade level of the reader. It ranges from 0 to 100. The higher the easier the text is to read.

$$Flesch = 206.835 - (1.015 \times \text{ASL}) - (84.6 \times \frac{n_{sy}}{n_w})$$

- **Scrabble (Scrabble Measure).** Represents the mean Scrabble letter values of all words.
- **FOG (Gunning's Fog Index [Gunning, 1952]).** The idea of this index is that short sentences written in plain English achieve a better score than long sentences written in complicated language. The ideal score for readability with the Fog index is 7 or 8. Anything above 12 is too hard for most people to read.

$$FOG = 0.4 \times (\text{ASL} + 100 \times \frac{n_{wsy\geq 3}}{n_w})$$

where $n_{wsy\geq 3}$ is the number of words with three syllables or more.

- **SMOG (SMOG (Regression Equation C) [Mc Laughlin, 1969]).** It estimates the years of education a person needs to understand a piece of writing.

$$SMOG.C = 0.9986 \times \sqrt{Nwmin3sy \times \frac{30}{n_{\text{st}}} + 5} + 2.8795$$

where $n_{wsy\geq 3}$ is the number of words with three syllables or more.

- **FORCAST (Simplified Version of FORCAST.RGL [Caylor and Sticht, 1973]).** It measures the grade level needed to read a text. It is designed to analyse technical text and it is considered ideal for multiple-choice tests, surveys and guides.

$$FORCAST = 20 - \frac{n_{wsy=1} \times 150}{(n_w \times 10)}$$

where $n_{wsy=1}$ is the number of one-syllable words.

3.3 Meta-feature definition

For meta-feature definition, we do not only search a set of characteristics that affect performance for any linguistic task, but we also want these features to have a scale.

In some cases, the scale is clear. For instance, the number of verbs, connectors, adverbs, etc. in a text (divided by the total number of words) is a proportion that goes from zero to one. However, defining a scale becomes more complicated for the non-propositional aspects mentioned in Section 2.2. For instance, what is the scale of negation? Is it just zero for no negation and one when there's any negation? For example, in the sentence:

Paula is not going to the concert and Maria is not going to the cinema.

We are sure about the number of nouns, it is four. So if we had to define the proportion of nouns it is 4/15. However, for the level of negation, we could define different metrics. We could describe negation as the existence of any negation, so this sentence would have level of negation 1. But we could also define negation as the number of words implying negativity, so this sentence would have level 2, because there are two "not".

In general, we want scales for which we can unambiguously assign a number to examples, but we have to admit that in many cases, this will be arbitrary. Also, in some cases, the result will not be the same if we calculate it at the granularity of full text, paragraph or sentence. And, of course, many of these meta-features will be language dependent.

In Table 3.5, we enumerate the list of selected meta-features with their possible levels and some examples for them. In some cases, there seems to be a procedure or a rule for calculating these levels, e.g., 'negation' or 'compositionality'. In others, however, the levels are more arbitrary and depend on the use of the examples as 'anchors', i.e., reference points for determining the level of a particular category.

We make this distinction because, for many interesting meta-features, manual annotation seems to be the only option available. For instance, why do the sentences "We have no clue about where it is" and "I'm not sure who will win the election" have uncertainty? A person could take these examples and others and still extrapolate relatively well to other instances (of course with a margin of error) without being able to use any clear rules. Moreover, it is not only that there are no rules but also that there is no ground truth in these levels. They are a convention.

3.4 Prompt design

As explained before, the idea is to use language models using few-shot learning to annotate millions of examples. The prompt is constructed with a context about the meta-features and their scale. This is not strictly necessary but is usually helpful. Then we have a series of examples, each with the given level for the meta-feature. Finally, the last sentence represents the example that we want to annotate.

Meta-features	Scale and Levels	Examples
Uncertainty	0: complete certainty, ... 10: complete uncertainty	"The cat is in the house": 1 "She might not do it again": 7 "He may come this afternoon": 3 "We have no clue about where it is": 8 "It is a fact that a square has four sides": 0 "It's impossible to know who will win the lottery": 10 "I'm not sure who will win the election": 8
Negation	0: no negation 1: simple negation 2: double negation 3: negation with quantification 4: very complex negation ...	"I'm a rich man" : 0 "She has never had a dog": 1 "It's untrue that all houses without windows do not have any light": 4 "I don't know what I don't know": 2 "The suspect is not in the house": 1 "The car has not been driven by anyone in the team": 3 "Never say never": 2
Time	0: no time expressions 1: simple temporal expressions 2: double temporal expressions 3: complex temporal expressions ...	"He came before noon": 1 "The house is blue" : 0 "There's a meeting every two weeks" : 3 "The train arrived ten minutes after the plane has left": 2
Space	0: no space relationships 1: simple spatial expressions 2: double spatial expressions 3: complex spatial expressions ...	"The pen was on the table": 1 "There's no room between the two cars": 2 "Tomorrow is a bank holiday": 0 "The lamp was hanging from two ropes, one attached to the ceiling and the other to the window": 5
Vocabulary	0..1: Normalised from some aggregate metric of the -log freq of words or something similar as in semantic complexity metrics.	"The ball is big": 0.1219 "Procrastination jeopardises excellence": 0.4235 "The boy must apologise": 0.198 "Ignoramus was an ultracrepidarian reposte" : 0.8324
Modality	0: no modality 1: simple modality 2: double modality ...	"The woman walked into a bar": 0 "The boy must apologise": 1 "The boy thinks we can't do it" : 3
Theory of Mind	0: no theory of mind 1: simple theory of mind 2: double theory of mind ...	"He came to the reception before noon": 0 "She didn't want to buy a car": 1 "The boy thinks we can't do it": 1 "The child feared his parents wanted to punish him": 2
Reasoning	0: no reasoning 1: simple reasoning 2: complex reasoning ...	"He tripped because of the step" : 1 "He came before noon with a bag full of presents": 0 "The grass was wet but it was sunny so someone must have watered the plant": 2
Compositionality	1...number of levels	"He came before noon": 0 "He came before she arrived": 1 "The man wearing the tall hat came before she arrived": 2 "He came before noon with a bag full of presents": 0.
Anaphora	0: no anaphora 1: simple (one possible referent) 2: complex (>1 possible referents) ...	"Kim thinks that he is clever": 1 "While Stuart was telling Susan the news, she laughed at him": 2
Noise	0...number of typos per character wrt to the original text with no typos	"The ball is big" : 0 "The bll isbige" : 3/13 "The boy bust apologise": 1/20

Table 3.5: Description of linguistic meta-features

Prompt template

"You are a helpful expert on linguistics. You must help me annotate the level of {META-FEATURE} of a given sentence/s. Note that {META-FEATURE DEFINITION}. I will first give you a few examples to illustrate it (as few-shot learning). Then you will have to determine the level of {META-FEATURE} for a new sentence/s on a scale from {META-FEATURE SCALE}.
{META-FEATURE EXAMPLES}
Sentence: {INSTANCE}
Level of {META-FEATURE}:"

The meta-feature scales and examples are the ones defined in Table 3.5. The definitions we have adopted for the meta-features are as follows:

- **Uncertainty.** Refers to epistemic situations involving imperfect or unknown information.
- **Negation.** Refers to a denial, contradiction, or negative statement.
- **Time.** A temporal expression in a text is a sequence of tokens (words, numbers and characters) that denote time, duration, or frequency.
- **Space.** A spatial expression in a text is a sequence of tokens (words, numbers and characters) relating to the position, area, and size of things.
- **Vocabulary.** The vocabulary level is measured by a normalised metric of the log freq of words
- **Modality.** Refers to a classification of propositions on the basis of whether they claim necessity or possibility or impossibility
- **Theory of Mind.** In psychology, theory of mind refers to the capacity to understand other people by ascribing mental states to them.
- **Reasoning.** Is the process of forming conclusions, judgments, or inferences from facts or premises
- **Compositionality.** In semantics, the principle of compositionality states that the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them.
- **Anaphora.** Is the use of a linguistic unit, such as a pronoun, to refer to the same person or object as another unit, usually a noun.
- **Noise.** Is the number of typos per character with respect to the original text with no typos.

This approach has some disadvantages. It can cost many tokens if we need to annotate millions of examples with multiple meta-features. Also, we need to detect and process those cases where the language model gives an answer that is not in the level range (postprocessing the output). Finally, not everything is automated, we had to create the pool of annotated examples in Table 3.5 to be used for this few-shot setting.

CHAPTER 4
Experimental Setting

In this chapter, we present the conducted experiments to investigate the differences between using linguistic meta-features and readability or lexical complexity metrics to predict task difficulty in AI benchmarks.

4.1 Difficulty prediction

First, we need to set the formula for the difficulty of an instance, which is basically the complement of the average correctness of the models considered (those selected from BIG-bench or HELM).

$$diff = 1 - \frac{\sum_{i=1}^{N} correct_i}{N}$$

Where $correct_i$ refers to the result of the model i for that instance (0: incorrect, 1: correct) and N is the number of models executed for that task.

Thus, the difficulty of an instance depends on the performance of the models evaluated in that task. A difficulty of zero indicates that all the models have been able to solve the tasks, therefore it is an easy task for the population of models assessed. On the contrary, if the difficulty is one, it means that no model has solved the question correctly, so it is a complicated question.

Once the "ground-truth" difficulties are calculated, the model we use for predicting difficulty from the instance metrics and/or meta-features is a Random Forest Regressor. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [Breiman, 2001]. We use Random Forest because the data has a non-linear shape, so a linear regression model failed to extract the non-linear features.

We use the Random Forest Regressor available in Scikit-learn[1] with the default parameters. The data is randomly split into 75% for training and 25% for testing the model.

For evaluating its predictions, we consider two metrics:

- **Test Score.** Coefficient of determination R^2 of the test prediction. The best possible score is 1 and a constant model that always predicts the same value independent of the input would get a score of 0.0.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

- **RMSE. (Root Mean Square Error)**. It measures the average distance between the values predicted by the model and the actual difficulty values. The lower, the better the model's predictions.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - O_i)^2}{n}}$$

where P_i represents the predicted value for an instance i, O_i the actual value and n the number of instances.

Note that we train a different model for each different task and each different setting (readability and lexical diversity metrics or linguistic meta-features). That lets us discern the behaviour of individual tasks depending on their nature and analyse their demands and their influence in predicting task difficulty.

4.2 Readability metrics

One part of the study is to evaluate the ability of existing human readability and lexical diversity metrics to predict the performance of language models.

To compute these metrics for each instance, as explained in Section 3.2, we use the QUANTEDA library. The text considered for the calculation is the result of concatenating the input question and the correct answer.

4.3 Meta-feature preparation

Regardless of their input structure, both HELM and BIG-bench task questions and options contain several sentences. In Table 3.5, we set the example for few-shot learning as sentences. As the examples given to the language model for annotation are individual phrases, we find it more convenient to separate the paragraphs of task questions into sentences too, as it has been proved to improve its performance.

For instance, if we have the paragraph:

"Question: A fire started in a forest but it wasn't started by people. What could have been the cause?
A. a careless bird
B. a smoking bear
C. electricity
D. a campfire
Answer: C"

We will query the LLM to get the meta-features for the following sentences:

- "Question: A fire started in a forest but it wasn't started by people."
- "What could have been the cause?"
- "A. a careless bird"
- "B. a smoking bear"
- "C. electricity"
- "D. a campfire"

Original Output Example	Corrected Value
N/A	0
-1	0
Infinity	5
It will rain tomorrow	0

Table 4.1: Examples of annotation outputs given by GPT-4 and their corresponding value after the postprocessing stage

- "Answer: C"

In order to achieve this division we use NLTK[2], which is a platform for building Python programs to work with human language data. It includes a tokenizer package, from which we use the sentence tokenizer. However, sentences like a "A. a careless bird" would be interpreted as two sentences "A." and "a careless bird", so these cases are handled separately.

While making this division into sentences, to distinguish the different types of phrases in future analysis, these are assigned an ID (1: question, 2: option, 3: right option). In this way, we have the opportunity to make difficulty predictions including different parts of the text. For example, we can predict the difficulty of the question plus the correct option, or of the question and the possible options, etc. Finally, the calculation of the meta-features for the whole questions (how the scores for all the sentences are combined) is explained in later sections.

4.4 Meta-feature annotation

For all the conducted experiments, to obtain the meta-feature values we query GPT-4 [OpenAI, 2023]. To set the "temperature" parameter we consider that higher values of this parameter make the output more random, while lower values make it more focused and deterministic. Hence, we set it to zero to make it as deterministic as possible, so we do not need to query the model again for repeated sentences, saving tokens and time.

4.5 Postprocessing

As explained in Section 3.4, one of the disadvantages of the annotation method is that we need to postprocess the output to ensure all the values are within the described scale, since the LLM sometimes gives a value that is not in the possible range of values.

To correct these values we follow the following criteria. Those values below the range are assigned zero, while those above are assigned the maximum value of the meta-feature scale. The language model sometimes replies with sentences that have nothing to do with the question, e.g. "Tomorrow will be Saturday", in this case, the output becomes zero. If there is a statement that it cannot answer, it is assigned to zero too. Table 4.1 lists the most representative cases.

For example, we have the following output from GPT-4 for this certain sentence:

[2]https://www.nltk.org/index.html

Sentence: *"Marcos shuld be in the lab because he has to finish his work today."*

Uncertainty: 4
Negation: N/A
Time: 1
Space: 1
Vocabulary: 0.121
Modality: 1
Theory of Mind: -1
Reasoning: 1
Compositionality: 1
Anaphora : 1
Noise: 1/54

After the postprocessing step we would obtain the following values:

Sentence: *"Marcos shuld be in the lab because he has to finish his work today."*

Uncertainty: 4
Negation: 0
Time: 1
Space: 1
Vocabulary: 0.121
Modality: 1
Theory of Mind: 0
Reasoning: 1
Compositionality: 1
Anaphora: 1
Noise: 0.019

Subsequently, once all the values are corrected (in Figure 4.1 and Figure 4.2 we can see the range of values obtained for each meta-feature for the tasks *BBQ* and *Epistemic Reasoning*), we can calculate the values of the meta-features for all the instances, i.e. for the whole paragraphs.

Finally, after exploring different options (mean, median, maximum), we consider that the level of a meta-feature of a whole instance is the mean of the levels of its sentences. That is because, for instance, we have a paragraph composed of three sentences.

> *Billy was so excited to win the race for class president. He worked so hard to win, and he was so proud of himself. His opponent wasn't so happy.*

It is clear that for "Time" meta-feature the paragraph value should be zero, there are no time expressions in any of the sentences.

However, for "Negation" is different, there is one sentence with negation. If we decided that the level of the paragraph is the maximum of all sentences, the level would be one. However, using the mean, we would obtain a "Negation" value of 0.33, which reveals that there is negation in that paragraph and that there is added difficulty, but not as great as if all the sentences had negation, a fact not reflected by the maximum.

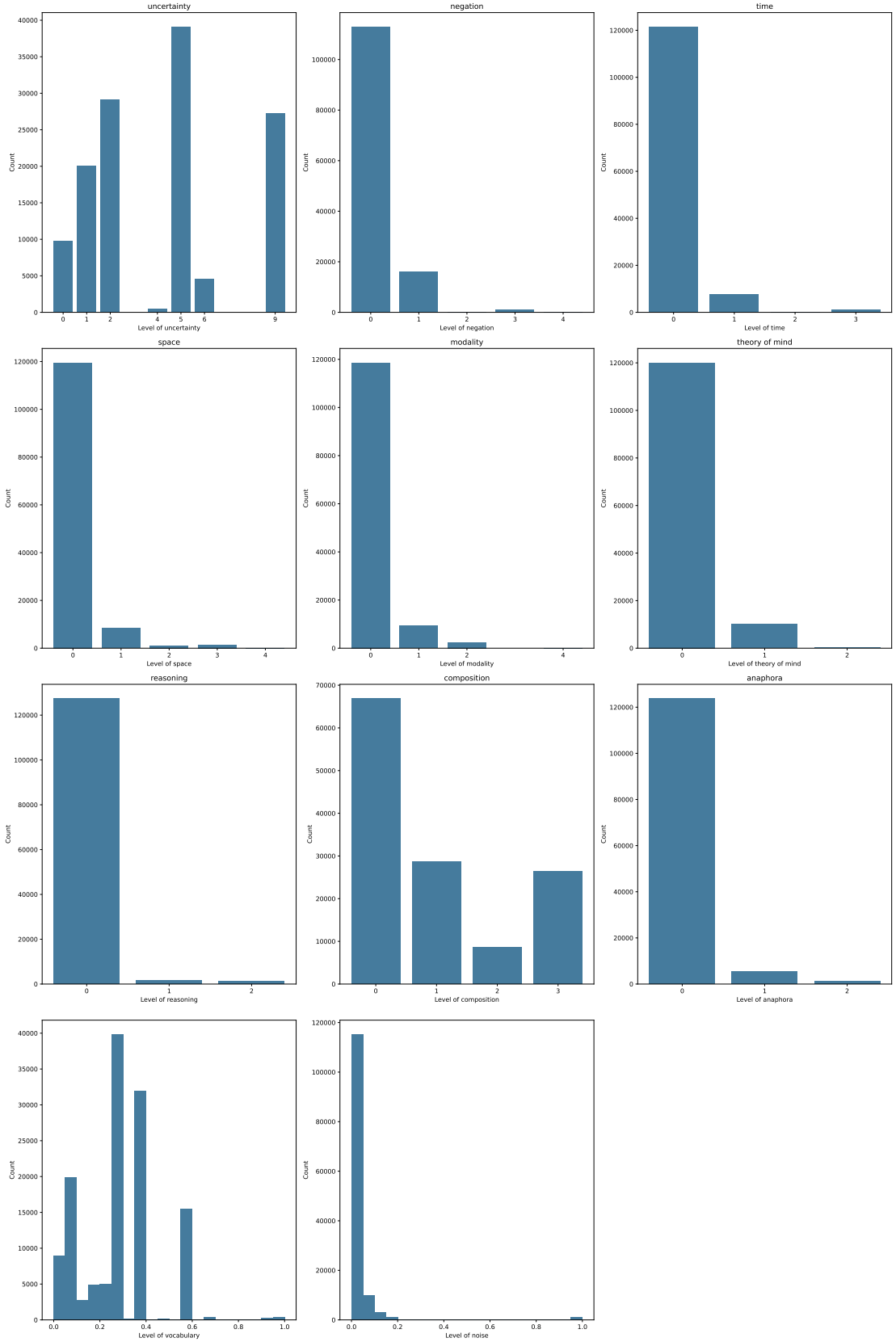

Figure 4.1: Resulting values for discrete meta-features after post-processing incorrect values of *BBQ*

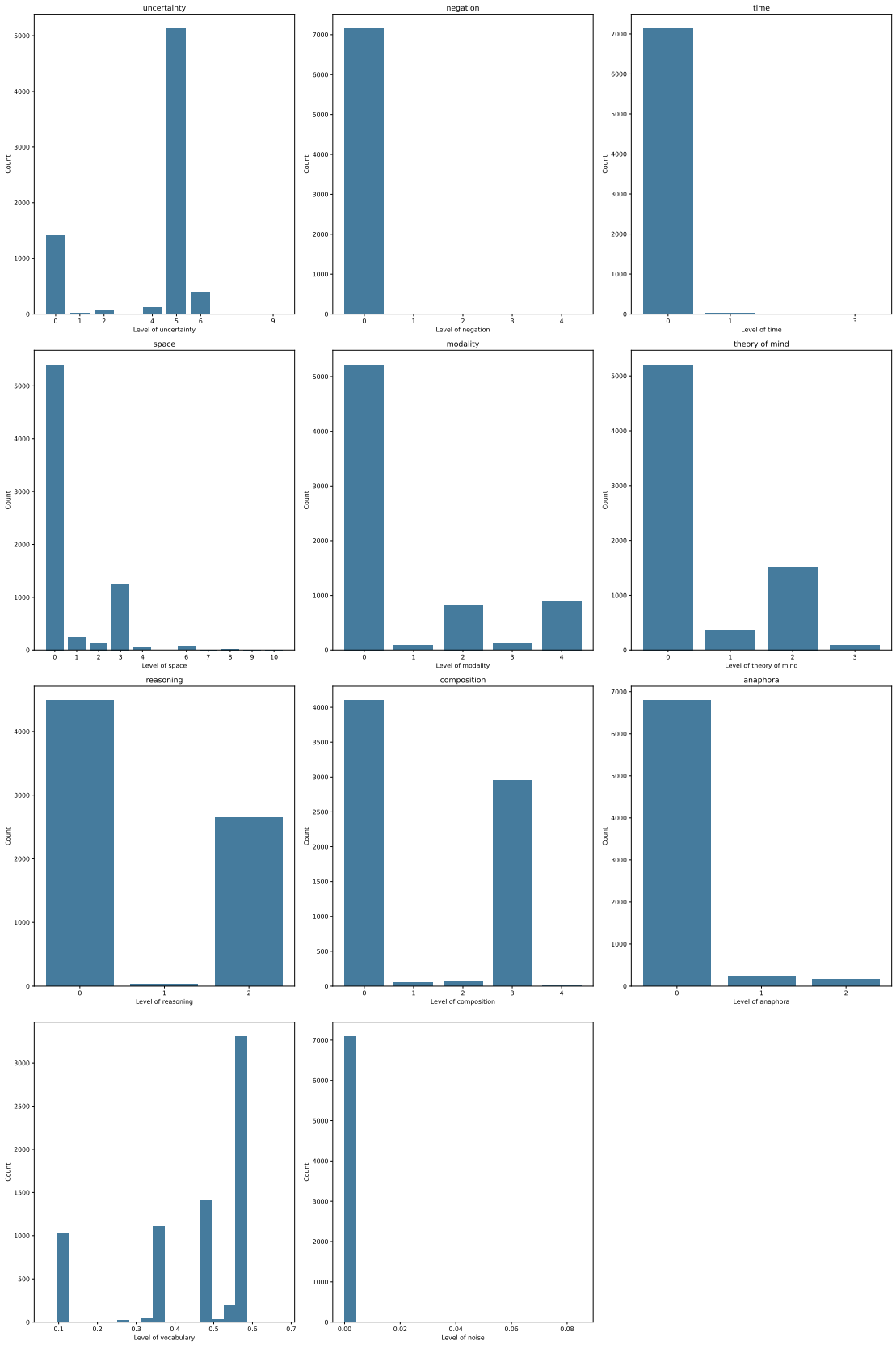

Figure 4.2: Resulting values for meta-features after post-processing incorrect values of *Epistemic Reasoning*

CHAPTER 5
Results

In this chapter, we analyse the results of the experiments, focusing on the differences between using lexical diversity and readability metrics or linguistic meta-features to predict task difficulty.

5.1 Overview

For a first general analysis, we focus on R^2, as it gives a good indicator of whether we can predict difficulty. At a first glance at Table 5.1, we can observe:

1. Tasks with high predictability using both linguistic meta-features and lexical/readability metrics.
2. Tasks that traditional metrics fail to predict difficulty but linguistic meta-features perform well.
3. Tasks where neither lexical/readability metrics nor meta-features properly predict difficulty.

The first group, which is composed of tasks with R^2 closer to 1, includes *BBQ*, *MMLU College Chemistry*, *MMLU Computer Security*, *MMLU Econometrics*, *MMLU US Foreign Policy* and *TruthfulQA*. In Figure 5.1, we can see how for instance, for *MMLU US Foreign Policy*, actual difficulty and predicted one are pretty similar. This result makes clear that these tasks have a linguistic base, and because of that, we can predict their difficulty both with linguistic meta-features and traditional metrics. The question of which of the two approaches is better will be addressed below.

Then there are those whose difficulty is not well predicted by traditional metrics, *Epistemic Reasoning* and *Formal Fallacies Syllogisms Negation*. We can take a look at Figure 5.2 and Figure 5.3 to see the difference between meta-features and readability metrics predictions in *Epistemic Reasoning*. Maybe for these tasks, linguistic meta-features are able to capture some relevant characteristics that traditional metrics do not. Yet an R^2 analysis is not enough to dive deeper into this matter. Later on, we will analyze other factors that may explain this fact.

Finally, there are some tasks where neither lexical/readability metrics nor meta-features properly predict difficulty. These are *Abstract Narrative Understanding*, *Hellaswag*, *LSAT*, *Legal Support* and *OpenbookQA*. In Figure 5.4 we observe the results of *LSAT*, the model predictions do not fit with the actual ones.

One of the reasons for this poor predictability seems to be the scarcity of evaluated models. Among the selected tasks, *Hellaswag* and *OpenbookQA* only have results for six

Task	Linguistic Meta-features	Traditional Metrics
Abstract Narrative Understanding	0.06	-0.01
BBQ	0.62	0.5
Epistemic Reasoning	0.9	-0.03
Formal Fallacies Syllogisms Negation	0.6	-0.15
Hellaswag	0.02	-0.03
Legal Support	0.3	0.05
LSAT	-0.07	-0.07
MMLU College Chemistry	0.77	0.74
MMLU Computer Security	0.83	0.85
MMLU Econometrics	0.68	0.7
MMLU US Foreign Policy	0.8	0.83
OpenbookQA	-0.04	0.01
TruthfulQA	0.59	0.56

Table 5.1: R^2 obtained in the test split when predicting difficulty with linguistic meta-features and lexical and readability metrics

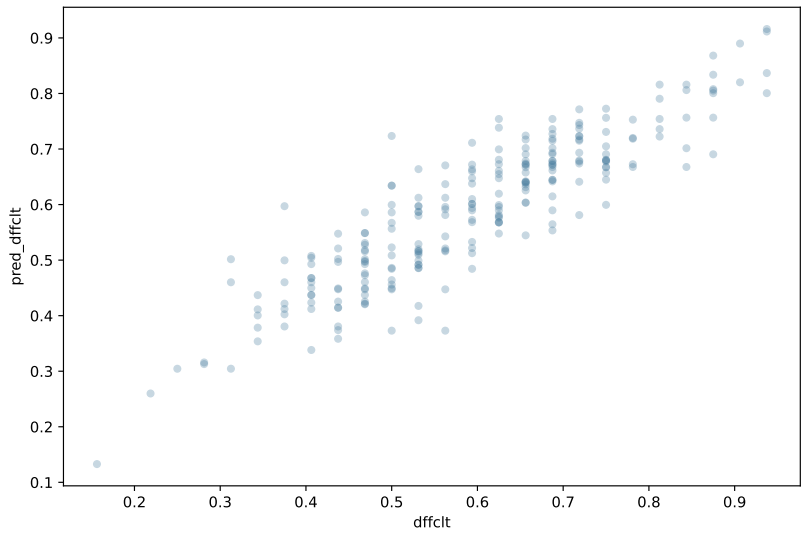

Figure 5.1: Predicted Difficulty vs. Actual Difficulty for *MMLU US Foreign Policy* using linguistic meta-features

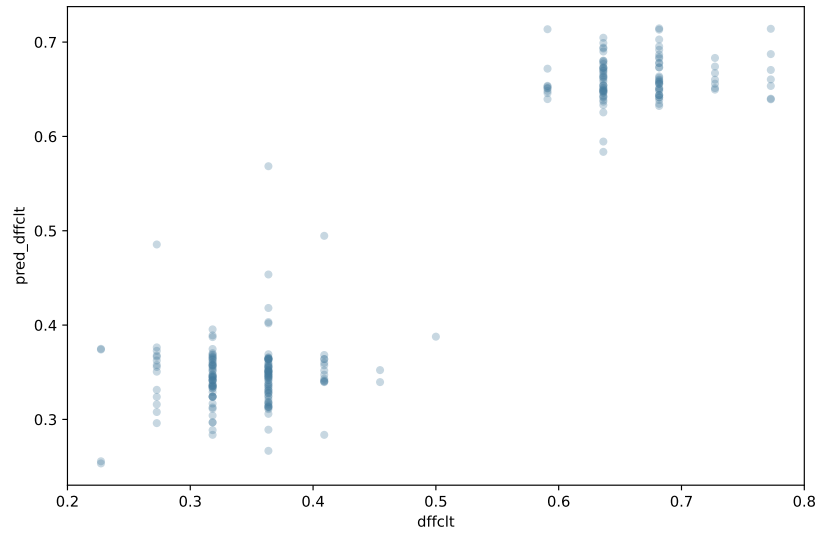

Figure 5.2: Predicted Difficulty vs. Actual Difficulty for *Epistemic Reasoning* using linguistic meta-features

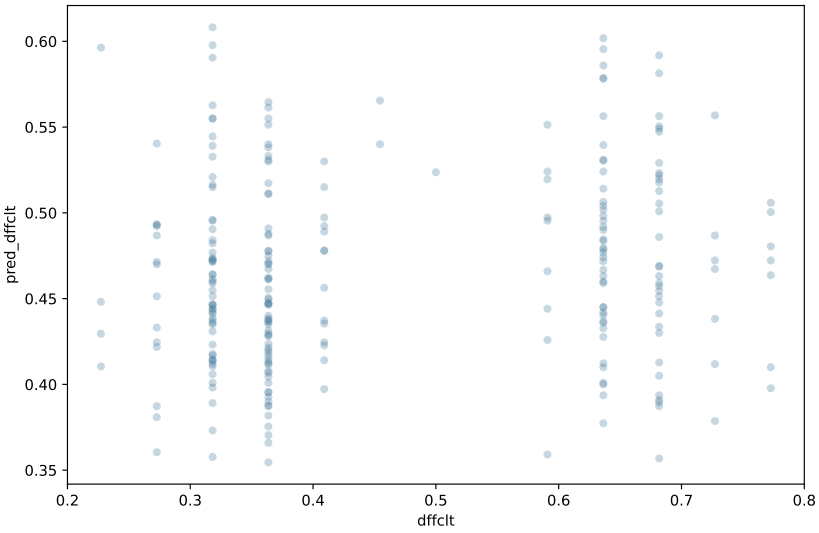

Figure 5.3: Predicted Difficulty vs. Actual Difficulty for *Epistemic Reasoning* using lexical and readability metrics

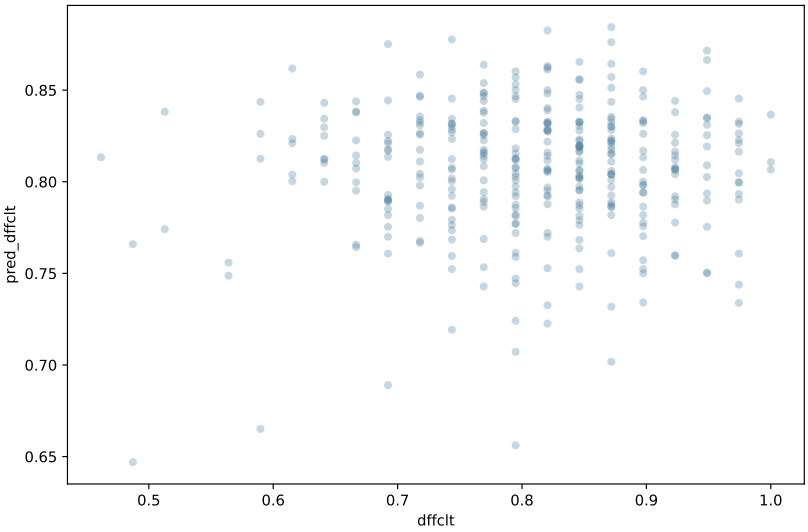

Figure 5.4: Predicted Difficulty vs. Actual Difficulty for *LSAT* using linguistic meta-features

models (unlike the rest which have at least 32), and this may influence the predictions. In Figure 5.5, the predicted difficulty for *OpenbookQA* is represented. As there are only six models, there are few difficulty values, so lines are formed perpendicular to the x-axis. However, the rest of the tasks with low R^2 do not show this characteristic. We did not find any other pattern in the data that could indicate the reason for this low predictability. Maybe these tasks simply do not have a linguistic basis.

In summary, we observe there are tasks that have a generic linguistic basis and others that do not, our approach helps to discern. Still, we cannot be sure that the ones that are not predictable have no linguistic basis, perhaps their difficulty is influenced by factors that neither linguistic meta-features nor lexical and readability metrics include.

After analyzing the R^2, there remains the doubt of which of the two options is better for predicting difficulty. When comparing the two main approaches, we can see that, on average, linguistic features have a lower RMSE (0.122) compared to readability and lexical complexity metrics (0.140). For most tasks, linguistic meta-features do a very good job predicting task difficulty.

All this suggests that **linguistic meta-features, labelled with GPT4, are effective to estimate task difficulty of other language models**. Also, **linguistic meta-features may provide a more accurate estimate of task difficulty than lexical diversity and readability metrics**.

However, on closer inspection, we observe several patterns in the behaviour of tasks with respect to their RMSE results. In Figure 5.6 we can see the RMSE for each of the tasks: the red bar represents the RMSE for lexical diversity and readability metrics, while the blue one shows the RMSE for linguistic meta-features. Despite the higher aggregate average, linguistic meta-features do not always predict better, we can discern three different behaviours with respect to their RMSE results. In general, we can classify tasks into these groups:

1. **Tasks with lower RMSE using linguistic meta-features.** These are *Abstract Narrative Understanding*, *BBQ*, *Epistemic Reasoning*, *Formal Fallacies Syllogisms Negation*, *Hellaswag*, *Legal Support*, *MMLU College Chemistry* and *TruthfulQA*.

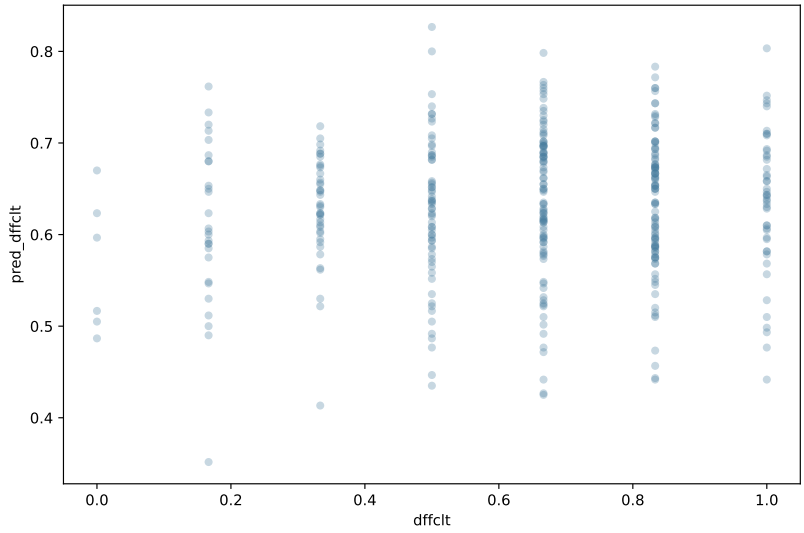

Figure 5.5: Predicted Difficulty vs. Actual Difficulty for *OpenbookQA* using linguistic meta-features

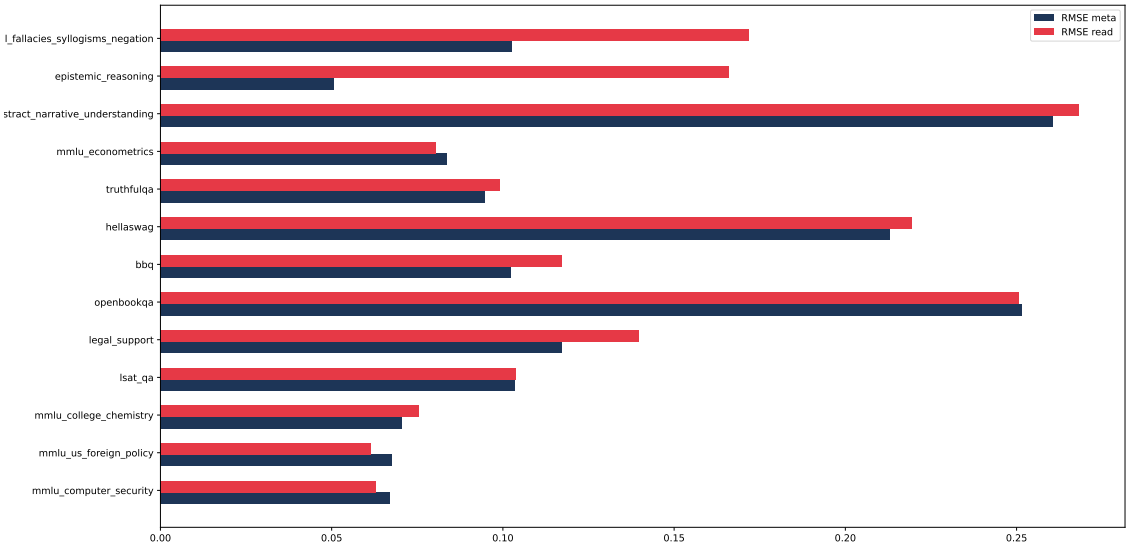

Figure 5.6: RMSE values obtained from predicting text difficulty for each task using linguistic meta-features and lexical diversity and readability metrics

Meta-feature	Average Rank	Metric	Average Rank
Vocabulary	1.46	Scrabble	2.31
Uncertainty	2.38	TTR	2.54
Noise	5.77	K	2.77
Compositionality	6	FORCAST.RGL	3.54

Table 5.2: Average Ranking Position in feature importance from the four best positioned meta-features and readability/lexical metrics

2. **Tasks with lower RMSE using lexical diversity and readability metrics.** *MMLU Computer Security*, *MMLU US Foreign Policy* and *MMLU Econometrics*.

3. **Tasks with similar RMSE for both feature sets.** Include *OpenbookQA* and *LSAT QA*.

But before diving deep into the different categories, it is relevant to note down what the majority of the tasks have in common. We gain valuable insights analysing feature importance, i.e. what features are most important for the model.

On the side of meta-features, vocabulary and uncertainty are valuable for all the tasks, we can observe (see Table 5.2) they have a higher average ranking in feature importance, 1.46 and 2.38 respectively. Related to lexical diversity and readability, Scrabble, TTR, and K are the most important for determining performance, with an average ranking of 2.3, 2.54 and 2.77, showing they are important for every task.

The fact that both TTR and K for the traditional metrics (both of lexical diversity) and vocabulary for the meta-features are significant features, may indicate that in the selected tasks the lexical component is considerably determinant of the difficulty of the tasks.

5.2 Tasks with lower RMSE using linguistic meta-features

This first category includes the tasks that show a more accurate estimation of difficulty when linguistic meta-features are used. These are *Abstract Narrative Understanding*, *BBQ*, *Epistemic Reasoning*, *Formal Fallacies Syllogisms Negation*, *Hellaswag*, *Legal Support*, *MMLU College Chemistry* and *TruthfulQA*.

In these tasks, theory of mind, reasoning and anaphora play a more significant role in contributing to the difficulty of the task than in other cases as seen in table 5.3. That may be because these tasks typically require higher-order cognitive abilities and may involve complex problem-solving, abstract reasoning or relational inference, which are effectively captured by linguistic meta-features. Let us take a closer look into the tasks of this category to prove this statement.

Most of the tasks in this group are classified as purely reasoning tasks as we can see in Table 3.1. *Abstract Narrative Understanding* involves analogical and social reasoning, and it shows in the feature importance ranking. Reasoning is the third one, only after vocabulary and uncertainty, meta-features shared by almost every task. Then both *Epistemic Reasoning* and *Formal Fallacies Syllogisms Negation* are logical reasoning tasks. Moreover, *Epistemic Reasoning* is specifically designed to test models' relationship to the theory of mind, so the fact that meta-features outperform readability metrics in capturing its difficulty is not unexpected. Here is a query of this task:

Meta-feature	Cat. 1	Cat. 2	Cat. 3
Uncertainty	2.12	2.67	3.00
Negation	7.50	3.33	10.50
Time	7.25	8.67	7.5
Space	7.00	6.67	5.00
Vocabulary	1.75	1.00	1.00
Modality	6.88	6.67	6.00
Theory of Mind	7.00	9.00	10.50
Reasoning	6.38	7.67	8.50
Compositionality	6.63	6.33	3.00
Anaphora	6.38	10.00	8.00
Noise	7.12	4.00	3.00

Table 5.3: Average ranking of each meta-feature in feature importance for each category

Identify the relation between the following premises and hypotheses, choosing from the options 'entailment' or 'non-entailment'.
Premise: Charles remembers that Amelia believes that a black scientist looks through a scope examining a biological specimen's blood cells.
Hypothesis: Amelia believes that a black scientist looks through a scope examining a biological specimen's blood cells.
Relation:

Besides reasoning, we have tasks categorised as knowledge question-answering, *Hellaswag*, *MMLU College Chemistry* and *TruthfulQA*, which a priori do not seem to have as clear demands of reasoning or theory of mind as the previous group. In fact, the feature importance of these factors is slightly lower. Moreover, the difference between lexical diversity and readability metrics in the RMSE is smaller than in pure reasoning tasks. However, if we analyse the task questions, we see they focus on commonsense knowledge. Here is an example from Hellaswag:

Question: Family Life: [header] How to stop your child from gossiping [title] Remind them that gossip can hurt feelings. [step] Bad news typically spreads faster than good news does, and gossip usually isn't good. Rumors are often made to hurt someone and their reputation.

A. You should teach your child that gossip is " inevitable " and will not go away once a few people talk about it. You might say, " if you and tommy start gossiping, your reputation will suffer.

B. Remind your child that gossiping in front of their friends can hurt any friendship or affection they have. Try to remind your child that gossip doesn't damage their reputation.

C. However, when someone's reputation becomes damaging or ugly, bad news can hurt other people's feelings. [substeps] Tell them that gossip keeps your child away from people that matter.

D. Let your child know that the subject of the rumor will likely hear it and probably become hurt. [substeps] Repeat back to your child the gossip they've spread and make it about them.

Answer:

This paragraph does not only require conceptual knowledge but needs a greater understanding of reality.

5.3 Tasks with lower RMSE using lexical diversity and readability metrics

The second category is composed of tasks that obtain lower RMSE when using lexical diversity and readability metrics to predict their difficulty. These are *MMLU Computer Security*, *MMLU US Foreign Policy* and *MMLU Econometrics*.

Tasks with lower RMSE when using readability metrics usually rely on simpler language processing, vocabulary and comprehension skills. For these tasks, readability metrics such as Flesch Reading Ease or FORCAST.RGL provide a more reliable estimate of difficulty, as they emphasise the complexity and readability of the text, which is directly related to the inherent difficulty of the tasks.

All the tasks belonging to this category are classified as knowledge question answering in Table 3.3. In fact, they are different domains of the same benchmark, Massive Multitask Language Understanding. These tasks require theoretical knowledge of their respective domains, but do not require complex reasoning.

This is an example of *MMLU Computer Security*.

Question: Exploitation of the Heartbleed bug permits
A. overwriting cryptographic keys in memory
B. a kind of code injection
C. a read outside bounds of a buffer
D. a format string attack
Answer: C

In order to answer this question you need to know the concept of "Hearbleed bug", but you do not need high cognitive skills.

It might be striking that all the MMLU tasks are in this group except for the College Chemistry domain. However, comparing their queries, we see that College Chemistry demands much more practical knowledge than the other domains.

To see the difference clearly, here is a question from US Foreign Policy domain

> Question: How did NSC-68 change U.S. strategy?
> A. It globalized containment.
> B. It militarized containment.
> C. It called for the development of the hydrogen bomb.
> D. All of the above
> Answer: D

And here is one from College Chemistry

> Question: A 0.217 g sample of HgO (molar mass = 217 g) reacts with excess iodide ions according to the reaction shown above. Titration of the resulting solution requires how many mL of 0.10 M HCl to reach equivalence point?
> A. 1.0 mL
> B. 10 mL
> C. 20 mL
> D. 50 mL
> Answer: C

Maybe not all the questions have this direct difference, but it is a factor that has affected the prediction of difficulty.

5.4 Tasks with similar RMSE for both feature sets

These tasks show roughly equal performance on both linguistic meta-features and lexical/readability metrics. Such tasks are *OpenbookQA* and *LSAT QA*.

In cases where tasks show similar RMSE results for both, it can be inferred that these tend to involve a combination of lower and higher order cognitive skills. The difficulty of such tasks does not derive solely from linguistic complexity or higher-order reasoning skills, but from a combination of both. If we take a closer look into *OpenbookQA* we can see these characteristics.

> Question: A fire started in a forest but it wasn't started by people. What could have been the cause?
> A. a careless bird
> B. a smoking bear
> C. electricity
> D. a campfire
> Answer: C

We observe that the question requires common sense, but it does not require very high level of reasoning as in tasks such as *Epistemic Reasoning*, so meta-features related to this capability are not as relevant as in the first category.

These benchmarks are characterised by having high compositionality and noise feature relevance.

CHAPTER 6
Conclusions and Future Work

The focus of this work was to demonstrate the general applicability of the meta-feature approach, its effectiveness and its value in assessing the complexity of different tasks.

The results in Chapter 5 show that we have defined a set of meta-features that are generally able to predict the difficulty of the series of tasks we have analysed. This allows us to know in a more detailed way the demands of each of the instances of a task and, more broadly, of the task itself.

But in addition to introducing this set of meta-features for tasks that use natural language, by obtaining these values that predict difficulty, we validated the annotation of the tasks using language models. We have been able to see that, despite the inconvenience of having to create some examples or to build a template, the automatic annotation of the instances does work and is much faster than if it had been done manually. It is true that we cannot be absolutely certain that every single instance has been annotated correctly, but this is not ensured when using human annotators either.

Overall, we have been able to work with up to 13 tasks, each of them with thousands of instances, many with several sentences, showing that this methodology is scalable to many other existing tasks.

The other important point to take away from this study is that, although meta-features obtained a lower RMSE than traditional readability metrics, the better or worse performance of one or the other depends on the tasks analysed. The classification of tasks into different groups based on their RMSE performance reveals specific characteristics of each task that pose unique challenges to difficulty estimation. This provides insights into the underlying linguistic and cognitive factors that influence task difficulty, and the methodological approaches needed to investigate them further. In other words, looking at feature relevance per task, we can know what are the elements that are really measured by the task.

It is thus clear from these observations that the effectiveness of linguistic meta-features and readability metrics varies depending on the nature of the task. Therefore, the optimal approach to modelling task difficulty may involve a selective combination of features that address the specific cognitive and linguistic demands of the given task.

This work has some limitations. Some of them are given by the breadth of the study. For reasons of time and compute, we could not explore all the tasks in BIG-bench and HELM, but this would be impractical for a project of this magnitude, our resources and time constraints. But this is definitely possible for others to do. Another limitation is the table of meta-features and their anchors that we used for the automated annotations. We made some choices there, but a more completely list of meta-features, and a possible taxonomy that is more strongly based on linguistics and cognitive science could lead to

better insights. Apart from not having the expertise in these areas, our goal was to show that the approach works, not to find the perfect rubric and the perfect anchors. In fact, because of this, we think that the results could only be better in the future.

There are many possibilities of future work, some derived form the limitations above. This work could be extended with more tasks and models, as they are available and the resources make it possible. It would also be interesting to explore the combination of linguistic meta-features and lexical/readability metrics depending on the characteristics of a given task.

It is important that for this future work to happen we leave the methodology, code and data available for others. First, with my supervisors, I am writing a paper out of this report and will submit it to a conference or a journal. Second, all the code and data can be in this Github repository[1], and other students or researchers can use it for their own studies.

On a personal level, by performing this project, I have been able to work with the latest AI technology such as language models. In particular, I have been working with GPT-4, which has been released very recently and is more powerful than the standard Chat-GPT, which everybody is talking about. As a computer scientist, I have gained a deeper understanding of how AI is evaluated and what are the latest trends in this field, which I would love to continue working on. The focus of this work on linguistics has also made me learn new concepts from another field that is related to computer science.

This study has also helped me to grow on a technical level, not only because I have become familiar with machine learning techniques such as Random Forest, completing my training in the branch of computing of my degree, but I would especially highlight the progress with data processing tools, both the collection of data and its pre-processing to perform experiments, obtaining results and their subsequent post-processing and visualisation. I think all of these skills will be extremely valuable in my future career as a computer scientist, in artificial intelligence or another field.

I would like to emphasise that the development of this thesis has required knowledge acquired in the degree of Computer Science. Subjects such as Statistics, Programming and Intelligent Systems have given me the foundations. And then, in the Computing branch, subjects such as Machine Learning, Techniques, Tools and Applications of Artificial Intelligence and Medical Informatics, have provided me with a solid knowledge in Artificial Intelligence. Special mention to Information Storage and Retrieval Systems, which has given me the NLP foundations necessary to write this study.

Finally, the work required the application of several transversal competencies. The most related were:

- **Analysis and Problem Solving.** One of the main challenges of this work was the large number of results obtained. It has required a great analysis capacity to find patterns and features that revealed the reasons for their behaviour, in order to draw valuable conclusions.
- **Creativity, Innovation and Entrepreneurship.** This study not only presents a new method to evaluate linguistic complexity but also uses large language models to automate the annotation process. The utilization of these models demonstrates the willingness to adopt and adapt new tools and technologies to address problems more efficiently and effectively. This requires a creative and innovative mindset.
- **Effective Communication.** This competency has been key to write this thesis. The proposal of novel methods in a field not everybody is familiar with, makes it es-

[1] https://github.com/yaelmd/TFG

sential to use clear and concise language, giving examples and adding graphic resources to convey the desired information to the reader.

- **Critical Thinking.** In order to carry out this project, it has been necessary to critically evaluate previously existing approaches to predict the difficulty of a text. Moreover, the proposed approach is compared with previous metrics, assessing impartially which metrics might be better depending on the task.

  In addition, I also recognise the advantages and disadvantages of the automated annotation method I propose. This shows a critical attitude towards my own approach and the ability to evaluate its implications and limitations. Critical thinking has enabled me to identify areas for improvement, consider possible biases and address them objectively.

To sum up, all the knowledge I have acquired during the Computer Science degree, both subjects and transversal competencies, have been really useful in facing the challenges that came up during this project, allowing me to meet the objectives set.

CHAPTER 7
Acknowledgments

Me gustaría dar las gracias al DMIP por acogerme este último año y medio con tanto cariño. He aprendido muchísimo de vosotros y me habéis ayudado a encontrar mi camino en el mundo de la informática, la investigación. Gracias por todas las oportunidades que me habéis dado.

En especial, quiero dar las gracias a mis tutores Jose y Nando, sin vosotros este trabajo no habría salido adelante. Gracias por guiarme durante este proceso, vuestros comentarios han sido muy valiosos. ¡Creo que por fin puedo decir que he aprendido a ilustrar! Para mí sois toda una referencia y espero seguir creciendo como investigadora gracias a vosotros. Mención especial a Lex, que ha sido un apoyo muy importante en el lab, ayudándome siempre que ha podido. Estoy segura de que te va a ir muy bien en tu próxima etapa.

También me gustaría recordar a mis compañeros de carrera, que se han convertido en compañeros de vida, que han hecho que las clases fueran mucho más amenas y los exámenes mucho menos dramáticos. Nunca voy a olvidar estar "in the easter monkey" con vosotros, a "volti", la paloma de la biblioteca, el parque de las colchonetas elásticas o "mundijuegos.com". Ha tenido que acabar la carrera para que deje de robaros patatas. Medalla al mérito a las personas que han sabido calmar mis nervios y me han hecho reir siempre que lo he necesitado, me llevo un recuerdo muy especial de vosotros.

Y como se suele decir, por último pero no por ello menos importante, a mi familia y a todas aquellas personas que forman parte de mi vida fuera de la universidad y me han apoyado para llegar hasta aquí. Os quiero mucho.

Bibliography

[Beyret et al., 2019] Beyret, B., Hernández-Orallo, J., Cheke, L., Halina, M., Shanahan, M., and Crosby, M. (2019). The animal-ai environment: Training and testing animal-like artificial cognition. *arXiv preprint arXiv:1909.07483*.

[Bostrom and Durrett, 2020] Bostrom, K. and Durrett, G. (2020). Byte pair encoding is suboptimal for language model pretraining. *arXiv preprint arXiv:2004.03720*.

[Bragg et al., 2021] Bragg, J., Cohan, A., Lo, K., and Beltagy, I. (2021). Flex: Unifying evaluation for few-shot nlp. *Advances in Neural Information Processing Systems*, 34:15787–15800.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

[Brown et al., 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[Burnell et al.,] Burnell, R., Burden, J., Rutar, D., Voudouris, K., Cheke, L., and Hernández-Orallo, J. Not a number: Identifying instance features for capability-oriented evaluation.

[Burnell et al., 2023a] Burnell, R., Schellaert, W., Burden, J., Ullman, T. D., Martinez-Plumed, F., Tenenbaum, J. B., Rutar, D., Cheke, L. G., Sohl-Dickstein, J., Mitchell, M., et al. (2023a). Rethink reporting of evaluation results in ai. *Science*, 380(6641):136–138.

[Burnell et al., 2023b] Burnell, R., Schellaert, W., Burden, J., Ullman, T. D., Martinez-Plumed, F., Tenenbaum, J. B., Rutar, D., Cheke, L. G., Sohl-Dickstein, J., Mitchell, M., Kiela, D., Shanahan, M., Voorhees, E. M., Cohn, A. G., Leibo, J. Z., and Hernandez-Orallo, J. (2023b). Rethink reporting of evaluation results in ai. *Science*, 380(6641):136–138.

[Caylor and Sticht, 1973] Caylor, J. S. and Sticht, T. G. (1973). Development of a simple readability index for job reading material.

[Chowdhary, 2020] Chowdhary, K. R. (2020). *Natural Language Processing*, pages 603–649. Springer India, New Delhi.

[Crosby et al., 2020] Crosby, M., Beyret, B., Shanahan, M., Hernández-Orallo, J., Cheke, L., and Halina, M. (2020). The animal-ai testbed and competition. In *Neurips 2019 competition and demonstration track*, pages 164–176. PMLR.

[Crossley et al., 2023] Crossley, S., Heintz, A., Choi, J. S., Batchelor, J., Karimi, M., and Malatinszky, A. (2023). A large-scaled corpus for assessing text readability. *Behavior Research Methods*, 55(2):491–507.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Dong et al., 2022] Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. (2022). A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

[Embretson and Reise, 2013] Embretson, S. E. and Reise, S. P. (2013). *Item response theory*. Psychology Press.

[Flesch, 1943] Flesch, R. (1943). Marks of readable style; a study in adult education. *Teachers College Contributions to Education*.

[François and Miltsakaki, 2012] François, T. and Miltsakaki, E. (2012). Do nlp and machine learning improve traditional readability formulas? In *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, pages 49–57.

[Graesser et al., 2011] Graesser, A. C., McNamara, D. S., and Kulikowich, J. M. (2011). Coh-metrix: Providing multilevel analyses of text characteristics. *Educational researcher*, 40(5):223–234.

[Gunning, 1952] Gunning, R. (1952). The technique of clear writing. mcgraw-hill. *New York*.

[He et al., 2023] He, X., Lin, Z., Gong, Y., Jin, A.-L., Zhang, H., Lin, C., Jiao, J., Yiu, S. M., Duan, N., and Chen, W. (2023). Annollm: Making large language models to be better crowdsourced annotators.

[Hendrycks et al., 2020] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. (2020). Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

[Hernández-Orallo, 2017] Hernández-Orallo, J. (2017). Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement. *Artificial Intelligence Review*, 48:397–447.

[Khurana et al., 2023] Khurana, D., Koli, A., Khatter, K., and Singh, S. (2023). Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, 82(3):3713–3744.

[Kintsch and Vipond, 2014] Kintsch, W. and Vipond, D. (2014). Reading comprehension and readability in educational practice and psychological theory. *Perspectives on learning and memory*, pages 329–365.

[Liang et al., 2022] Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C. D., Ré, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., Wang, J., Santhanam, K., Orr, L., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N., Guha, N., Chatterji, N., Khattab, O., Henderson, P., Huang, Q., Chi, R., Xie, S. M., Santurkar, S., Ganguli, S., Hashimoto, T., Icard, T., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., and Koreeda, Y. (2022). Holistic evaluation of language models.

[Mahowald et al., 2023] Mahowald, K., Ivanova, A. A., Blank, I. A., Kanwisher, N., Tenenbaum, J. B., and Fedorenko, E. (2023). Dissociating language and thought in large language models: a cognitive perspective. *arXiv preprint arXiv:2301.06627*.

[Manning and Schutze, 1999] Manning, C. and Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

[Martínez-Plumed et al., 2016] Martínez-Plumed, F., Prudêncio, R. B., Martínez-Usó, A., and Hernández-Orallo, J. (2016). Making sense of item response theory in machine learning. In *ECAI 2016*, pages 1140–1148. IOS Press.

[Martínez-Plumed et al., 2019] Martínez-Plumed, F., Prudêncio, R. B., Martínez-Usó, A., and Hernández-Orallo, J. (2019). Item response theory in ai: Analysing machine learning classifiers at the instance level. *Artificial intelligence*, 271:18–42.

[Mc Laughlin, 1969] Mc Laughlin, G. H. (1969). Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

[Morante and Sporleder, 2012] Morante, R. and Sporleder, C. (2012). Modality and negation: An introduction to the special issue. *Computational linguistics*, 38(2):223–260.

[OpenAI, 2023] OpenAI (2023). Gpt-4 technical report.

[Srivastava et al., 2023] Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., Lewkowycz, A., Agarwal, A., Power, A., Ray, A., Warstadt, A., Kocurek, A. W., Safaya, A., Tazarv, A., Xiang, A., Parrish, A., Nie, A., Hussain, A., Askell, A., Dsouza, A., Slone, A., Rahane, A., Iyer, A. S., Andreassen, A., Madotto, A., Santilli, A., Stuhlmüller, A., Dai, A., La, A., Lampinen, A., Zou, A., Jiang, A., Chen, A., Vuong, A., Gupta, A., Gottardi, A., Norelli, A., Venkatesh, A., Gholamidavoodi, A., Tabassum, A., Menezes, A., Kirubarajan, A., Mullokandov, A., Sabharwal, A., Herrick, A., Efrat, A., Erdem, A., Karakaş, A., Roberts, B. R., Loe, B. S., Zoph, B., Bojanowski, B., Özyurt, B., Hedayatnia, B., Neyshabur, B., Inden, B., Stein, B., Ekmekci, B., Lin, B. Y., Howald, B., Orinion, B., Diao, C., Dour, C., Stinson, C., Argueta, C., Ramírez, C. F., Singh, C., Rathkopf, C., Meng, C., Baral, C., Wu, C., Callison-Burch, C., Waites, C., Voigt, C., Manning, C. D., Potts, C., Ramirez, C., Rivera, C. E., Siro, C., Raffel, C., Ashcraft, C., Garbacea, C., Sileo, D., Garrette, D., Hendrycks, D., Kilman, D., Roth, D., Freeman, D., Khashabi, D., Levy, D., González, D. M., Perszyk, D., Hernandez, D., Chen, D., Ippolito, D., Gilboa, D., Dohan, D., Drakard, D., Jurgens, D., Datta, D., Ganguli, D., Emelin, D., Kleyko, D., Yuret, D., Chen, D., Tam, D., Hupkes, D., Misra, D., Buzan, D., Mollo, D. C., Yang, D., Lee, D.-H., Schrader, D., Shutova, E., Cubuk, E. D., Segal, E., Hagerman, E., Barnes, E., Donoway, E., Pavlick, E., Rodola, E., Lam, E., Chu, E., Tang, E., Erdem, E., Chang, E., Chi, E. A., Dyer, E., Jerzak, E., Kim, E., Manyasi, E. E., Zheltonozhskii, E., Xia, F., Siar, F., Martínez-Plumed, F., Happé, F., Chollet, F., Rong, F., Mishra, G., Winata, G. I., de Melo, G., Kruszewski, G., Parascandolo, G., Mariani, G., Wang, G., Jaimovitch-López, G., Betz, G., Gur-Ari, G., Galijasevic, H., Kim, H., Rashkin, H., Hajishirzi, H., Mehta, H., Bogar, H., Shevlin, H., Schütze, H., Yakura, H., Zhang, H., Wong, H. M., Ng, I., Noble, I., Jumelet, J., Geissinger, J., Kernion, J., Hilton, J., Lee, J., Fisac, J. F., Simon, J. B., Koppel, J., Zheng, J., Zou, J., Kocoń, J., Thompson, J., Wingfield, J., Kaplan, J., Radom, J., Sohl-Dickstein, J., Phang, J., Wei, J., Yosinski, J., Novikova, J., Bosscher, J., Marsh, J., Kim, J., Taal, J., Engel, J., Alabi, J., Xu, J., Song, J., Tang, J., Waweru, J., Burden, J., Miller, J., Balis, J. U., Batchelder, J., Berant, J., Frohberg, J., Rozen, J., Hernandez-Orallo, J., Boudeman, J., Guerr, J., Jones, J., Tenenbaum, J. B., Rule, J. S., Chua, J., Kanclerz, K., Livescu, K., Krauth, K., Gopalakrishnan, K., Ignatyeva, K., Markert, K., Dhole, K. D., Gimpel, K., Omondi, K., Mathewson, K., Chiafullo, K., Shkaruta, K., Shridhar, K., McDonell, K., Richardson, K., Reynolds, L., Gao, L., Zhang, L., Dugan, L., Qin, L., Contreras-Ochando, L., Morency, L.-P., Moschella, L., Lam, L., Noble, L., Schmidt, L., He, L., Colón, L. O., Metz, L., Şenel, L. K., Bosma, M.,

Sap, M., ter Hoeve, M., Farooqi, M., Faruqui, M., Mazeika, M., Baturan, M., Marelli, M., Maru, M., Quintana, M. J. R., Tolkiehn, M., Giulianelli, M., Lewis, M., Potthast, M., Leavitt, M. L., Hagen, M., Schubert, M., Baitemirova, M. O., Arnaud, M., McElrath, M., Yee, M. A., Cohen, M., Gu, M., Ivanitskiy, M., Starritt, M., Strube, M., Swędrowski, M., Bevilacqua, M., Yasunaga, M., Kale, M., Cain, M., Xu, M., Suzgun, M., Walker, M., Tiwari, M., Bansal, M., Aminnaseri, M., Geva, M., Gheini, M., T, M. V., Peng, N., Chi, N. A., Lee, N., Krakover, N. G.-A., Cameron, N., Roberts, N., Doiron, N., Martinez, N., Nangia, N., Deckers, N., Muennighoff, N., Keskar, N. S., Iyer, N. S., Constant, N., Fiedel, N., Wen, N., Zhang, O., Agha, O., Elbaghdadi, O., Levy, O., Evans, O., Casares, P. A. M., Doshi, P., Fung, P., Liang, P. P., Vicol, P., Alipoormolabashi, P., Liao, P., Liang, P., Chang, P., Eckersley, P., Htut, P. M., Hwang, P., Miłkowski, P., Patil, P., Pezeshkpour, P., Oli, P., Mei, Q., Lyu, Q., Chen, Q., Banjade, R., Rudolph, R. E., Gabriel, R., Habacker, R., Risco, R., Millière, R., Garg, R., Barnes, R., Saurous, R. A., Arakawa, R., Raymaekers, R., Frank, R., Sikand, R., Novak, R., Sitelew, R., LeBras, R., Liu, R., Jacobs, R., Zhang, R., Salakhutdinov, R., Chi, R., Lee, R., Stovall, R., Teehan, R., Yang, R., Singh, S., Mohammad, S. M., Anand, S., Dillavou, S., Shleifer, S., Wiseman, S., Gruetter, S., Bowman, S. R., Schoenholz, S. S., Han, S., Kwatra, S., Rous, S. A., Ghazarian, S., Ghosh, S., Casey, S., Bischoff, S., Gehrmann, S., Schuster, S., Sadeghi, S., Hamdan, S., Zhou, S., Srivastava, S., Shi, S., Singh, S., Asaadi, S., Gu, S. S., Pachchigar, S., Toshniwal, S., Upadhyay, S., Shyamolima, Debnath, Shakeri, S., Thormeyer, S., Melzi, S., Reddy, S., Makini, S. P., Lee, S.-H., Torene, S., Hatwar, S., Dehaene, S., Divic, S., Ermon, S., Biderman, S., Lin, S., Prasad, S., Piantadosi, S. T., Shieber, S. M., Misherghi, S., Kiritchenko, S., Mishra, S., Linzen, T., Schuster, T., Li, T., Yu, T., Ali, T., Hashimoto, T., Wu, T.-L., Desbordes, T., Rothschild, T., Phan, T., Wang, T., Nkinyili, T., Schick, T., Kornev, T., Tunduny, T., Gerstenberg, T., Chang, T., Neeraj, T., Khot, T., Shultz, T., Shaham, U., Misra, V., Demberg, V., Nyamai, V., Raunak, V., Ramasesh, V., Prabhu, V. U., Padmakumar, V., Srikumar, V., Fedus, W., Saunders, W., Zhang, W., Vossen, W., Ren, X., Tong, X., Zhao, X., Wu, X., Shen, X., Yaghoobzadeh, Y., Lakretz, Y., Song, Y., Bahri, Y., Choi, Y., Yang, Y., Hao, Y., Chen, Y., Belinkov, Y., Hou, Y., Hou, Y., Bai, Y., Seid, Z., Zhao, Z., Wang, Z., Wang, Z. J., Wang, Z., and Wu, Z. (2023). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

[Thoppilan et al., 2022] Thoppilan, R., Freitas, D. D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., Li, Y., Lee, H., Zheng, H. S., Ghafouri, A., Menegali, M., Huang, Y., Krikun, M., Lepikhin, D., Qin, J., Chen, D., Xu, Y., Chen, Z., Roberts, A., Bosma, M., Zhao, V., Zhou, Y., Chang, C.-C., Krivokon, I., Rusch, W., Pickett, M., Srinivasan, P., Man, L., Meier-Hellstern, K., Morris, M. R., Doshi, T., Santos, R. D., Duke, T., Soraker, J., Zevenbergen, B., Prabhakaran, V., Diaz, M., Hutchinson, B., Olson, K., Molina, A., Hoffman-John, E., Lee, J., Aroyo, L., Rajakumar, R., Butryna, A., Lamm, M., Kuzmina, V., Fenton, J., Cohen, A., Bernstein, R., Kurzweil, R., Aguera-Arcas, B., Cui, C., Croak, M., Chi, E., and Le, Q. (2022). Lamda: Language models for dialog applications.

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

[Tweedie and Baayen, 1998] Tweedie, F. J. and Baayen, R. H. (1998). How variable may a constant be? measures of lexical richness in perspective. *Computers and the Humanities*, 32:323–352.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Vilar and Federico, 2021] Vilar, D. and Federico, M. (2021). A statistical extension of byte-pair encoding. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 263–275.

[Voudouris et al., 2022] Voudouris, K., Donnelly, N., Rutar, D., Burnell, R., Burden, J., Hernández-Orallo, J., and Cheke, L. G. (2022). Evaluating object permanence in embodied agents using the animal-ai environment. *https://ceur-ws. org/Vol-3169/paper2. pdf*.

APPENDIX A
Sustainable Development Goals

In 2015, the United Nations adopted the 2030 Agenda for Sustainable Development. This agenda defines 17 universally applicable Sustainable Development Goals (SDGs) to push economic growth, commitment to social needs and protection of the environment. These SDGs relate to this project as listed in the following table.

Sustainable Development Goals	**High**	**Medium**	**Low**	**Not applicable**
SDG 1. **No poverty.**				X
SDG 2. **Zero hunger.**				X
SDG 3. **Good health and well-being.**				X
SDG 4. **Quality education.**		X		
SDG 5. **Gender equality.**				X
SDG 6. **Clean water and sanitation.**				X
SDG 7. **Affordable and clean energy.**				X
SDG 8. **Decent work and economic growth.**				X
SDG 9. **Industry, innovation and infrastructure.**	X			
SDG 10. **Reduced inequalities.**				X
SDG 11. **Sustainable cities and communities.**				X
SDG 12. **Responsible consumption and production.**				X
SDG 13. **Climate action.**				X
SDG 14. **Life below water.**				X
SDG 15. **Life on land.**				X
SDG 16. **Peace, justice and strong institutions.**			X	
SDG 17. **Partnership for the goals.**		X		

The Sustainable Development Goal which is clearly related to this work, is number 9, Industry, Innovation and Infrastructure. The project proposes an innovative method for predicting difficulty in NLP tasks. This implies progress in the field of Artificial Intelligence evaluation, understanding better system capabilities and promoting their improvement. Besides, the presentation of an automated approach for annotating the meta-features entails a more efficient way of annotation, improving productivity in example creation. In addition, the exploration of automated processes aligns with the idea of sustainable industrialisation, potentially allowing researchers to reduce the dependency on manual annotation processes. This is translated to major efficiency and less resource consumption.

The project highlights that task-oriented evaluation makes no sense with the rise of general-purpose systems, as it has low predictability on how the systems will perform in new tasks. The proposed meta-feature approach means an improvement in evaluating these systems and, therefore, in increasing their reliability. This directly affects to SDG number 4, Quality Education. Today's extension of language models could mean a powerful tool for improving education, but it is indispensable it is reliable and transmits truthful information. Progress in the evaluation methods of these systems is key for achieving this goal.

Growth in general-purpose systems also aligns with SDG 16, Peace, Justice and Strong Institutions. Understanding better model capabilities and improving their evaluating methods promotes transparency, safety and the disappearance of biases in them, something essential to bring on trust in technology and make it fair.

Finally, in this work, we use collaborative repositories such as BIG-Bench or HELM to analyse the predictability of linguistic meta-features. These collaborations between researchers and organisations being publicly available, as well as the conduct of this type of research, encourage sharing data and relevant knowledge, proving a joint effort to improve Artificial Intelligence systems.

APPENDIX B
Code and data

Here we explain the major technologies involved in the code and a short introduction to it and how to process the data.

All the data manipulation in this work is done with Pandas[1], which is a fast, powerful, flexible and easy-to-use open-source data analysis and manipulation tool, built on top of Python. The AI Benchmarks used for this study, which are explained in detail in Section 3.1, store their data in JSON files. So we had to create a JSON parser that extracted the relevant features of these files. This is an example of the structure of a BIG-bench JSON File:

```
{
"model":{
"additional_details": ,
    "decoding_params":,
    "description":,
    "flop_matched_non_embedding_params":,
    "model_family":,
    "model_name":,
    "non_embedding_params":,
    "total_params":,
    "training_batch_size":,
    "training_steps":
},
"queries":[{
    "samples":[
        "absolute_scores": ,
                "correct": ,
                "input": ,
                "metrics": ,
                "normalized_scores":,
                "scores":,
                "target_values":,
                "targets":
    ]
    "shots":
    "task":
}]
}
```

[1] https://pandas.pydata.org/

Note that, for instance, for HELM tasks, we could know the response of the language model and the correct answer to a question, but it was not written explicitly if the system had given the right answer or not, so we needed to organise and calculate new data to fit in our objectives.

Each of the evaluated models has its own JSON File with the results for each instance of a task. Therefore, to facilitate the analysis of the data, we store the information of all the models for a task in a single Pandas dataframe, whose content is explained in Section 3.1.

Moreover, this work includes plots for visualising the results of our analysis. This is done with Matplotlib[2], a library for creating visualizations in Python. The instance level results for each task are stored in JSON files (one for each model evaluated). To facilitate the analysis of the data, we store the relevant features in a Pandas dataframe.

[2]https://matplotlib.org/