Hochschule Luzern - Informatik
Bachelor of Science in Artificial Intelligence & Machine Learning

Can Language Models Follow Discussions?

Fall Semester 2023
Date of Submission 03. January 2024

Author:
Nico Previtali

Advisors:
Andreas Waldis
Markus Weiler

Bachelor Thesis at Lucerne University of Applied Sciences and Arts
School of Computer Science and Information Technolog

Title of Bachelor Thesis: Can Language Models Follow Discussions?

Name of Student: Previtali Nico

Degree Program: BSc AIML

Year of Graduation: 2024

Main Advisor: Waldis , Andreas

External Expert: Weiler , Markus

Industry partner/provider: HSLU

Code / Thesis Classification:
☒ Public (Standard)
☐ Private

Declaration
I hereby declare that I have completed this thesis alone and without any unauthorized or external help. I further declare that all the sources, references, literature and any other associated resources have been correctly and appropriately cited and referenced. The confidentiality of the project provider (industry partner) as well as the intellectual property rights of the Lucerne University of Applied Sciences and Arts have been fully and entirely respected in completion of this thesis.

Place / Date, Signature Zürich, 03.01.2024, N. Previtali

Submission of the Thesis to the Portfolio Database:
Confirmation by the student
I hereby confirm that this bachelor thesis has been correctly uploaded to the Portfolio Database in line with the code of practice of the University. I rescind all responsibility and authorization after upload so that no changes or amendments to the document may be undertaken.

Place / Date, Signature Zürich, 03.01.2024, N. Previtali

Expression of thanks and gratitute

I would like to express my sincere gratitude to my main advisor, Andreas Waldis, for his invaluable guidance and support throughout my bachelor's program. His expertise and encouragement helped me to complete this research and write this thesis.

Abstract

Exploring the ability of Large Language Models (LLMs) to engage in human discussion is a novel challenge in Natural Language Processing (NLP). This work takes an innovative, iterative approach to designing probing tasks that assess LLMs' interpretive functions concerning human discussion dynamics, particularly developing linguistic tasks such as stance alignment and interactive dynamics informed by observed essential properties for discussion tracking. Using a dataset of structured debates from Kialo, this study critically evaluates models from three LLM families - GPT, BERT, and ELECTRA. It examines the effectiveness of these models in deciphering discourse and reflects on the creative process required to translate complex discourse concepts into actionable evaluation criteria. The findings highlight the impact of model architectures and training data on the performance of LLMs and draw attention to the current interpretability of these AI systems in the context of human-machine interaction. Through this in-depth analysis, the research uncovers different levels of LLM capabilities in simulating human-like discussion comprehension, making a meaningful contribution to understanding AI's communicative capabilities.

I find that the outcome largely varies for different model types, sizes, and datasets and is subject to significant variance concerning Probing formulation. The iterative Probing process leaves you open to creativity and flexibility, which unsettles you but simultaneously moves you and keeps your curiosity alive. Ultimately, this work feels more like a foundation that gives you the tools to verify the next generation of LLM models and move on with various new data sources. And to do so, you will enjoy this new AI adventure as I did.

Table of Contents

1 Introduction

1.1 Problem

Recent advances in Large Language Models (LLMs), highlighted by the emergence of systems such as ChatGPT, have broadened the scope of discussions to include human-machine interactions. This evolution in natural language processing represents a shift towards more sophisticated, interactive AI systems. However, a critical challenge in this field is the focus on delivering fast and accurate results at the expense of interpretability. The training of LLMs on petabyte-scale data, facilitated by rapid technological advances, often prioritises the result over the clarity of the underlying processes. In addition, there is a notable historical tendency, exemplified by the ELIZA program (a therapist chatbot developed around 1966), for humans to ascribe human-like qualities to machines. This tendency can lead to misconceptions about the true capabilities of AI.

1.2 The Guiding Questions

Can Language Models Follow Discussions? It remains to be seen to what extent different LLMs have already learned the skills needed to follow human discussions and which of their inherent properties (e.g., architecture, training data) help them to do so. Therefore, the guiding questions for this thesis are the following:

- Which characteristics are essential to be able to follow discussions?
- How can these properties be verified using Probing Tasks within Language Models?
- How do different language models differ on these tasks, and which properties are crucial for differences?
- What implicit understanding of a discussion can we expect from a language model?

1.3 Vision

This thesis investigates the ability of Large Language Models (LLMs) to understand complex human discourse. Data will be sourced from Kialo to create probing exercises that emulate a test scenario and allow an assessment of the interpretive functions of LLMs. For example, Probing Tasks may include analysing the stance of claims in political discussions to reveal degrees of agreement or disagreement.

Given the inherent complexity of human conversations and their representation in a dataset, this thesis acknowledges that it can only reveal a fraction of the whole landscape. This limitation is due to the one-semester timeframe, the multifaceted nature of discourse, and the limited scope of existing datasets. As such, this work is only a tiny endeavor in the broad field of understanding how LLMs comprehend and participate in human-like discourse.

While focused on applied engineering analysis, the results of this study aim to inform aspects of model interpretability and provide insights into ethically informed AI design. It also seeks to engage with the broader conversation about how society interacts with AI, which could notify considerations for future regulation.

2 State of research / Technology

This chapter lists and explains state of the art background knowledge relevant to understand this thesis.

2.1 Linguistic Theories

Two influential theories have emerged as linguistic pillars to understand the complexities of human discourse: Discourse Representation Theory (DRT) and Rhetorical Structure Theory (RST). These frameworks have provided foundational insights into underlying structural rhetoric and semantics.

2.1.1 Discourse Representation Theory (DRT)

Developed by Hans Kamp, DRT is a framework primarily aimed at formally representing the meaning of discourse. It utilizes Discourse Representation Structures (DRS) to model the mental constructs associated with discourse processing. DRT is particularly noted for its ability to handle meaning that spans across sentences and expresses temporal relations within discourse. This capability makes it suitable for modeling the evolution of argumentative lines and other complex discourse phenomena. The intricate handling of contextual dependencies and co-references allows DRT to analyse various nuanced aspects of language comprehension and production, such as anaphora resolution and presupposition. This theory has been summarized from *Handbook of Philosophical Logic* (Kamp et al., 2010).

2.1.2 Rhetorical Structure Theory (RST)

RST, formulated by William Mann and Sandra Thompson, concentrates on the rhetorical components of structuring and organizing texts. This theory is a fundamental tool for analysing syntactic and semantic attributes of text structure aimed at achieving rhetorical impact. RST

delves into how argumentative structures are composed within texts, providing methods to identify core arguments and their supportive satellite information. It explores the intricate relations between text segments, where each unit may serve a specific function (elaboration, contrast, cause, etc) and is bound together to create coherent and persuasive discourse. This theory has been summarized from of *Natural Language Generation* (Bateman & Delin, 2006).

2.2 Large language models

The following introduces the transformers concept, which is the basic architecture that all LLMs in this thesis use.

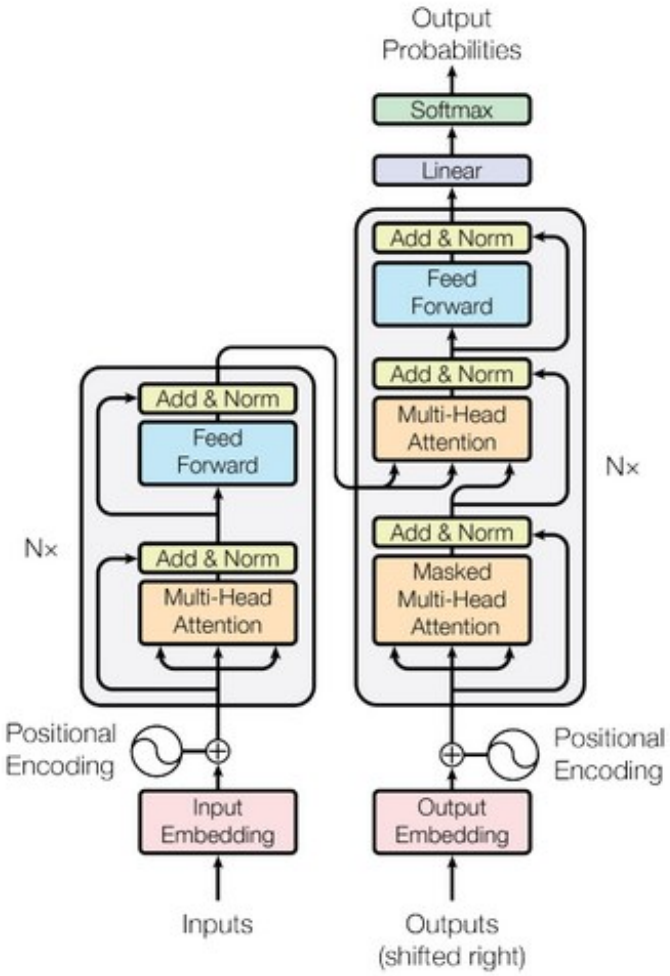

Figure 1 - Transformers Architecture

2.2.1 Transformers

Transformers, introduced in *Attention Is All You Need* (Vaswani et al., 2017), marked a significant shift in natural language processing. Their parallel processing capabilities, a departure from the sequential processing of earlier models like Recurrent Neural Networks (RNNs) and Long Short Term Memory Networks (LSTMs), have been pivotal in the evolution of language models.

2.2.2 Self-Attention

The self-attention mechanism, a defining feature of transformers, enables the model to contextually weigh the importance of different words in a sentence. This mechanism allows transformers to handle longer text sequences by focusing on relevant input parts, thereby facilitating the development of Large Language Models (LLMs) capable of understanding and generating human-like text.

For example, in the sentence "The cat sat on the mat," the model discerns that "cat" is closely related to "sat" and "mat."

2.2.3 Encoder-Decoder

- **Encoder-decoder architectures** are prevalent in NLP, where the encoder transforms the input text into a high-dimensional representation, capturing complex relationships between words and phrases. This representation, often called the "contextual representation," provides the decoder with a comprehensive understanding of the input text, enabling it to generate meaningful and coherent output. The high-dimensional nature of this representation allows NLP models to identify subtle semantic and syntactic relationships, surpassing the limitations of traditional models that rely on simpler representations.

- **Encoder-Focused Models** like BERT excel in tasks requiring a deep understanding of input text, analysing, and encoding information for comprehending context and relationships.
- **Decoder-Focused Models** such as GPT specialize in generating text predicting the next word in a sequence, adequate for tasks involving content creation and dialogue systems.

Model Variants

For this thesis, the LLM's three main variants include the following:

- **GPT** (Generative Pretrained Transformer) is trained to predict the next word in a sequence. This capability makes it practical for generating coherent and contextually relevant text. (*OpenAI GPT2 — Transformers 3.0.2 Documentation*, n.d.)
- **BERT** (Bidirectional Encoder Representations from Transformers) utilizes bidirectional context, analysing words within the context of their surrounding text. This approach is particularly beneficial for tasks where understanding the full context of a sentence is crucial. (*BERT — Transformers 3.0.2 Documentation*, n.d.)
- **ELECTRA-related Models (DeBERTa v3):** Uses an more advanced pre-training approach using a generator-discriminator framework. Unlike traditional Masked Language Modeling (MLM) methods such as BERT, which mask tokens and train the model to predict their original identities, ELECTRA corrupts the input by replacing some tokens with plausible alternatives from a generator network. The discriminator is then tasked with detecting which tokens have been replaced. This replaced token detection method has been shown to be more sampling efficient than MLM because it is designed to consider all input tokens rather than a small masked subset. Consequently, by learning from a more comprehensive input, ELECTRA's discriminative model often outperforms

BERT-like architectures, given the same model size, data and computational resources. (*ELECTRA, Transformers 3.0.2 Documentation*, n.d.)

2.2.4 Training

LLMs' performance and capabilities are significantly influenced by their training data and trained on vast and diverse datasets; the data's quality, diversity, and representativeness are crucial in shaping the models' understanding of language and biases as explored *UL2: Unifying Language Learning Paradigms* by (Tay et al., 2023). The learning process involves adjusting internal parameters to minimize the difference between the model's output and the expected result, directly impacting their ability to understand context, generate coherent responses, and perform various language tasks.

2.3 Capabilities and limitations of selected LLMs

The following studies have been collected from various sources focusing on LLMs capabilities and limitations which provide a backbone for assumptions and interpretations throughout this project.

2.3.1 Text length: BERT vs BART

Exploring the limitations of language models in processing long texts, the study *Towards Understanding Large-Scale Discourse Structures in Pre-Trained and Fine-Tuned Language Models* (Huber & Carenini, 2022) researched how models like BERT and BART handle extended discourse. For this thesis, the following study aspects are particularly interesting:

- **Advanced discourse extraction** using the sliding window approach to overcome the input-length limitations of traditional transformer-based models: This allows for broader contextual understanding.

- **Theoretical Alignment** with established discourse theories is demonstrated, suggesting a comprehensive grasp of complex discourse elements.
- **Performance Edge** is evident as these models show enhanced capabilities in discourse analysis, outperforming simpler counterparts.
- **Handling Long Documents** remains a challenge for BERT and BART, as they are constrained by their fixed input size, limiting their ability to capture and follow extended discourse fully.
- **Methodological Dependence** indicates that standard pre-trained models may struggle with large-scale discourse analysis without the sliding context window adaptation to handle long documents.

Implications for Discourse Tracking:

- The findings from this study highlight that while pre-trained models like BERT and BART are adept at understanding complex discourse structures, their effectiveness diminishes with the length of the discourse. BART can potentially encode more discourse structures than BERT due to its bigger attention matrix, which captures these features. However, evaluation shows that BERT outperforms BART in direct comparison, whereas fine-tuning improves BART´s performance and worsens BERT´s performance. This difference has been attributed to the models varying encoding input sizes (512 for BERT and 1024 for BART): The more words (tokens) can be processed at a time, the likelier it is for the model to learn long-distance semantic relationships between those words.
- Findings also show that similar discourse relationships are captured in the same heads across different fine-tuning tasks.

2.3.2 *Text length: DiscoBERT*

The paper *Discourse-Aware Neural Extractive Text Summarization* (Xu, Gan, Cheng, & Liu, 2019) shows a new approach to mitigate one of BERTs essential limitations: Since BERT had been trained on a token-level basis and not on a document-level basis, it does not capture long-range dependencies very well and often returns redundant or uninformative summarizations based on its limited contextual learning.

- **Textual Granularity:** Enhanced textual understanding is achieved by breaking down text into sub-sentential units (Elementary Discourse Units) rather than whole sentences. This allows the model to build relationships between smaller units, reducing superficial learning and improving summary conciseness.
- **Long-Range Dependencies:** RST-based discourse graphs enable the model to process extensive contextual relations, mitigating BERT's limitations in summarizing longer text as these graphs reveal underlying semantic structures.
- **Computational Intensity:** The structural analysis of a text is algorithmically demanding, requiring significant resources, which still constitute a limitation in large-scale applications.
- **External Tool Reliance:** DiscoBERT's performance is contingent upon the precision of parsing tools that determine the text structure, emphasizing its reliance on the accuracy of preliminary discourse analysis.
- **Handling Long Texts** remains a challenge for DiscoBERT, similar to BERT and BART, especially with long documents.

2.3.3 Training data

The study *UL2: Unifying Language Learning Paradigms* by (Tay et al., 2023) focuses mainly on the dependency of LLMs on the quality of pre-training data, a key limitation affecting their performances:

- **Data Dependence:** UL2 accentuates the profound impact of pre-training data on LLMs' performance, which extends to their proficiency in discourse comprehension. Quality and diversity of training sets are paramount for nuanced understanding.
- **Model Scalability:** While models like UL2 show potential in adaptability, the vast amounts of data needed for effective pre-training underscore the resource-intensive nature of these LLMs.
- **Training Data Limitations:** The scope and characteristics of the training corpus directly inform the LLM's ability to generalize and interpret complex discussion threads, highlighting a fundamental constraint in applying LLMs to real-world conversational contexts.

2.3.4 BERT's knowledge

The capabilities and limitations of BERT have been extensively explored in the study *A Primer in BERTology: What We Know About How BERT Works* (Rogers et al., 2020). This study has provided a comprehensive analysis of BERT's syntactic and semantic knowledge capabilities:

- **Hierarchical Structure** in BERT's representations indicates an ability to capture underlying syntactic tree structures.
- **Syntactic Information in Embeddings** is evident as BERT's token embeddings can recover syntactic trees. However, they encounter challenges with labels of distant parent

nodes (long-range dependencies), limiting their capability to understand broader contextual relationships.

- **Self-Attention Weights and Syntax** do not directly encode syntactic structure, making full parse tree extraction from BERT heads complex.

Semantic Knowledge

- **Understanding Semantic Roles** is a strength of BERT, as it demonstrates knowledge of semantic roles and shows preferences for semantically related fillers.
- **Entity Types and Relations** are encoded by BERT, detectable through probing classifiers, contributing to its ability to understand complex language structures.
- **Struggle with Numbers** reveals BERT's limitations in forming robust representations for numbers and in tasks involving numerical generalization.
- **Named Entity Recognition** is where BERT excels, but it shows brittleness to named entity replacements, suggesting a superficial understanding of named entities.

World Knowledge

- **Commonsense Knowledge** is a challenging area for BERT, as it needs to work on pragmatic inference and understand abstract object attributes.
- **Knowledge Induction via MLM** in BERT shows promise in certain relation types but is limited in reasoning based on world knowledge.

Limitations

- **Complexity of Probes** used with BERT raises questions about the reliance on the original model versus the probe's capabilities.
- **Contradictory Conclusions** can arise from different probing methods, leading to complementary or contradictory insights about BERT's knowledge capabilities.

2.3.5 HuggingFace Summary of selected LLMs

The following table shows the attributes to the respective LLMs named on HuggingFace´s model cards for each mentioned LLM.

Feature / Model	**microsoft deberta-v3-base**	**facebook bart-base**	**GPT-2**	**albert-base-v2**	**bert-base uncased**
Training Data Details	Diverse web text, BookCorpus	English web text, diverse sources	Web pages, fiction books	Unpublished books, Wikipedia (no lists, tables, headers)	Unpublished books, Wikipedia (no lists, tables, headers)
Data Indicative Learning	Broad linguistic understanding, context-aware predictions	Comprehensive language generation and comprehension	General language understanding, creative text generation	Broad language understanding, efficient language processing	Contextual understanding of language, relationship between sentences
Architecture Type	Transformer	Transformer (seq2seq)	Transformer	Transformer	Transformer
Layers	12 layers	Encoder: 6, Decoder: 6	48 layers	12 layers (repeating)	12 layers
Parameters	86 million	N/A (similar to BERT-base)	1.5 billion	11 million	110 million
Key Features	Gradient Disentangled Embedding Sharing	Bidirectional Encoder, Autoregressive Decoder	Autoregressive, Large-scale training	Parameter sharing, Embedding Factorization	Bidirectional context, Masked LM, NSP
Strengths	Enhanced performance on NLP tasks, Efficiency	Effective text generation and comprehension	Versatile task performance	Memory efficiency, Various task suitability	Contextual understanding, Versatility
Limitations	Computational cost similar to BERT	Less suited for non-generation tasks	Incoherence in long text passages	Less suited for text generation	Not ideal for text generation, distant node ranges
Favoured Tasks	NLP tasks like question answering, text classification	Text generation, Summarization, Translation	Translation, Summarization, Question answering	Classification, Token classification, QA	Sequence classification, Token classification, Q

2.3.6 Training Data and Learning of selected LLMs

The diversity and quality of training data profoundly influence LLMs' performance. For instance, models trained on more diverse and extensive datasets show a better grasp of complex discourse elements and context. The following section provides an overview of the for this project relevant LLMs training data used summarized from the official Hugginface website´s model cards and this Chapters previously mentioned studies.

DeBERTa v3 (microsoft/deberta-v3-base)

- **Training Data**: DeBERTa v3 was trained using a substantial dataset of 160GB, which included diverse web texts and BookCorpus. This dataset size provided the model with extensive exposure to various linguistic contexts.
- **Methodology and Features**: It uses ELECTRA-Style pre-training combined with Gradient Disentangled Embedding Sharing. This approach enhances the model's efficiency in NLU tasks. The model also employs an enhanced mask decoder to predict masked tokens during training better.
- **Performance and Application**: DeBERTa v3 significantly improved over its predecessors in downstream tasks like SQuAD 2.0 and MNLI, indicating its strong capabilities in question answering and textual entailment. *Microsoft/Deberta-V3-Base · Hugging Face* (n.d.)

BART (facebook/bart-base)

- **Training Data**: BART was primarily trained on English language texts, but the exact dataset details are less explicitly mentioned than for DeBERTa v3.
- **Methodology and Features**: BART is a transformer encoder-decoder (seq2seq) model with a bidirectional encoder (similar to BERT) and an autoregressive decoder (similar to

GPT). Its pre-training involved corrupting text with arbitrary noising functions and learning to reconstruct the original text. This included techniques like randomly shuffling sentences and replacing text spans with a single mask token.

- **Performance and Application**: BART excels in text generation tasks such as summarization and translation but also performs well in comprehension tasks like text classification and question answering. It matches the performance of models like RoBERTa in specific benchmarks, indicating its versatility. (*Albert-Base-v2 · Hugging Face*, n.d.)

GPT-2 (gpt2)

- **Training Data**: GPT-2 was pre-trained on BookCorpus, a dataset of over 7,000 self-published fiction books across various genres, and a dataset of 8 million web pages, known as the WebText dataset. This dataset was created by scraping web pages linked to Reddit posts with at least three upvotes before December 2017, ensuring a diverse range of internet text sources. Wikipedia pages were excluded to avoid overfitting.
- **Architecture and Features**: GPT-2 follows a transformer architecture with 1.5 billion parameters. It has several modifications from its predecessor, like moving layer normalization to the input of each sub-block and expanding the vocabulary to 50,257 tokens. The model also increased the context size from 512 to 1024 tokens and used a larger batch size.
- **Performance and Application**: Known for its ability to predict the next item in a sequence accurately, GPT-2 is adept at tasks like text translation, question answering, summarization, and text generation. However, it can generate repetitive or nonsensical text over longer passages. (*OpenAI GPT2 — Transformers 3.0.2 Documentation*, n.d.)

ALBERT (albert-base-v2)

- **Training Data**: Similar to BERT, ALBERT was trained on a combination of BookCorpus and English Wikipedia. The dataset provides a diverse and comprehensive foundation for language understanding.
- **Architecture and Features**: ALBERT is designed to reduce model size and improve training efficiency compared to BERT. It employs two parameter-reduction techniques: parameter-sharing across all layers and factorizing the embedding layer. This results in a smaller memory footprint and a model with fewer parameters.
- **Performance and Application**: ALBERT is suitable for tasks requiring an understanding of entire sentences or texts, such as sequence classification, token classification, and question answering. (*Albert-Base-v2 · Hugging Face*, n.d.)

BERT (bert-base-uncased)

- **Training Data**: BERT was pre-trained on BookCorpus and English Wikipedia. The dataset provides a vast range of text, contributing to a comprehensive language understanding.
- **Architecture and Features**: BERT is known for its masked language modeling (MLM) and next sentence prediction (NSP) pretraining tasks. The architecture is based on a transformer model, and it is designed to understand the context and relationships within and between sentences.
- **Performance and Application**: BERT is primarily used for tasks like sequence classification, token classification, or question answering. However, it is less suited for text-generation tasks. (*BERT — Transformers 3.0.2 Documentation*, n.d.)

2.4 Similar Studies

This subsection shows a selection of studies that have explored similar themes related to LLMs probing linguistic tasks.

2.4.1 "Am I the Bad One"?

The paper" Am *I the Bad One"? Predicting the Moral Judgement of the Crowd Using Pre-trained Language Models* (Alhassan et al., 2022) explores pre-trained LLMs' ability to predict moral judgments made by users of the specific Reddit forum "Am I the A*******?" (AITA). The study establishes the models' proficiency in capturing more profound concepts of meaning grounded in actions or societal context, indicating their potential to comprehend complex discussion scenarios. RoBERTa and Longformer achieved the best moral judgment prediction, with RoBERTa showing a strong correlation with the crowd's verdicts. Reddit as a data source is noted for its authentic texts, naturally offering a richness and complexity that is difficult to emulate in controlled datasets.

- **Method**: Fine-tuning of pre-trained language models tailored to the moral judgment prediction task.
- **Models Examined**: BERT, RoBERTa, RoBERTa-large, ALBERT, and Longformer.
- Dataset: A crowdfunded moral judgment dataset was constructed by scraping relevant data from the AITA subreddit—emphasizing the moral judgment of diverse daily situations and personal behaviors.

Dataset	Model	Sequence Length	Batch Size	Training Accuracy	Validation Accuracy	MCC	Confusion Matrix			
							TP	FP	TN	FN
Dataset1	BERT	512	8	0.78	0.78	0.091	203	174	21480	5823
	RoBERTA	512	8	0.78	0.78	**0.098**	131	52	21584	5913
	ALBERT	512	8	0.78	0.78	0	0	0	21584	6096
	Longformer	1024	4	0.78	0.78	0	0	0	21636	6044
Subset2	BERT	512	8	0.83	0.78	0.59	4007	583	5679	2019
	RoBERTA	512	8	0.81	**0.81**	**0.644**	4029	298	5912	2049
	RoBERTA Large	512	8	0.86	0.79	0.6	3882	440	5775	2191
		512	4	0.75	0.77	0.54	4308	1071	5144	1765
	ALBERT	512	8	0.83	0.80	0.623	3704	257	6093	2234
		512	16	0.76	0.79	0.62	3480	92	6258	2458
Subset3	RoBERTA	512	8	0.86	0.87	0.76	4527	63	5964	1492
	Longformer	1024	4	0.87	**0.88**	**0.77**	4698	140	5854	1354
		2048	2	0.87	0.87	0.763	4495	42	5985	1524

Figure 2 - Findings "Am I the Bad One"?

2.4.2 Discourse Probing

The study *Discourse Probing of Pretrained Language Models* (Koto et al., 2021) introduces document-level discourse probing to examine and compare the capabilities of various pre-trained language models' in capturing document-level discourse relationships. The findings suggest that all models performed well on simple tasks like next sentence prediction (NSP) but struggled with more complex discourse tasks such as sentence ordering—indicative of challenges in modelling multi-sentence discourse. BART performed best in its encoder layers, while BERT was noted for doing surprisingly well, especially in its deeper layers. Notably, there was significant variability in which layers and to what extent different models encoded discourse information, again highlighting influences of training data and objectives on LLMs' capabilities.

- **Method**: The study utilized various probing tasks, including next sentence prediction, sentence ordering, discourse connective prediction, RST relation prediction, etc.
- **Models Examined**: BART, BERT, RoBERTa, ALBERT, ELECTRA, GPT-2, and T5.
- **Dataset**: Tasks derived from sources like XSUM, Wikipedia, and various discourse treebanks tailored to English, Mandarin Chinese, German, and Spanish, focusing on document-level discourse relationships.

The study concludes that while models like BART and RoBERTa, which are trained on more data, tend to capture discourse better, nuances depend on the specific layer and design of the model. Models designed primarily for generation tasks, like T5 and GPT-2, show limitations in discourse understanding, while BART and RoBERTa's encoder layers excel in capturing discourse structure. The study has also found that deeper layers of these models tend to be where more sophisticated discourse knowledge is encoded.

2.4.3 Italian Transformer Probing

The study *Probing Linguistic Knowledge in Italian Neural Language Models across Language Varieties* (Alessio Miaschi et al., 2022) investigates the linguistic knowledge encoded in Italian pre-trained transformer models and identifies how the complexity of probing classifiers affects their ability to decipher the information within these representations. By analysing the performance of seven different Italian Neural Language Models (NLMs) across a spectrum of linguistic features, the research provides insights into the variations in implicit knowledge as influenced by text genres and language varieties. The analysis uses probing tasks derived from the Italian Universal Dependency Treebank (IUDT) to explore the capabilities of models such as multilingual BERT, language-specific BERT variants, RoBERTa models, and a GPT-2-based

model, "GePpeTto". The study evaluates these models using two probing classifiers: the simpler LinearSVR and the more complex MLP.

In particular, the results suggest that the complexity of the probing classifier can significantly affect the extraction of linguistic features from the learned representations of the models. In addition, it appears that the encoding of linguistic knowledge by Italian NLMs is not uniform across models, reflecting different training results depending on the text genre and language variety.

- **Method**: Probing with LinearSVR and MLP architectures to detect different linguistic properties and compare classifier´s complexities.
- **Models Examined**: Seven Italian NLMs, including multilingual BERT, language-specific BERT variations, and RoBERTa variations, as well as a GPT-2-based mode named “GePpeTto“.
- **Dataset**: The probing tasks used the Italian Universal Dependency Treebank (IUDT), covering a wide range of textual genres and language varieties.

Findings show the averaged results for the 2 different probing architectures:

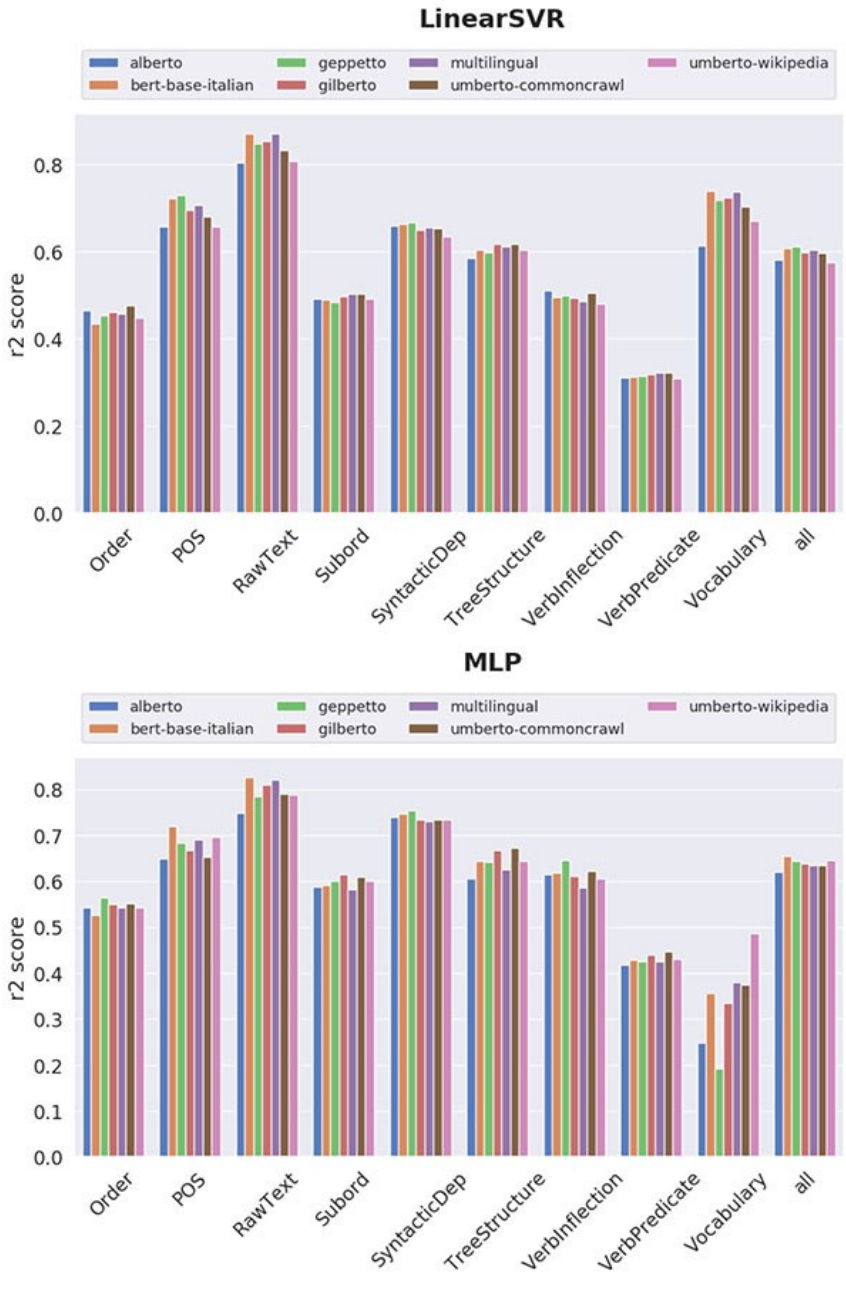

Figure 3 - Results "Italian Transformer Probing"

2.5 Model Interpretability

This subchapter introduces the concepts of the most common methods used for comparing LLMs.

2.5.1 Probing

Probing is a method to investigate whether a language model's encoder has learned and can effectively represent specific linguistic or informational patterns. By adding a simple linear classifier onto the model's encoder output, the probing method evaluates the ability of these internal representations to assist in classification tasks.

The general process consists of the following steps:

1. **Define the probe task:** Choose a simple linguistic task for the probe. For example, identifying Parts of Speech (POS) such as nouns, verbs or adjectives provides a clear focus for assessment.
2. **Collect training data** that include a variety of parts of speech examples, ensuring diverse exercises for the model to discern.
3. **Train classifier:** Construct a basic classifier on the selected linguistic task and data to interpret the encoder's representations. For example, the classifier is trained to identify parts of speech from the encoder's output.
4. **Isolate LLMs encoder:** Separate the encoder from the entire language model framework to concentrate exclusively on the representations it has learned.
5. **Run examples through encoders (forward pass):** Run the selected example sentences through the encoder, outputting representations that will be predicted by the probing classifier.

6. **Control task comparison:** Compare classifier performance with control tasks, such as random guessing, to ensure the model truly recognizes linguistic patterns and does not operate on chance.
7. **Evaluation:** Review performance metrics to conclude the encoder's effectiveness in capturing the designated linguistic task.

Control Tasks used in this project include the following:

Random Baseline: This task assesses if the model solves the task only using its learned knowledge and not the probing classifier. Even though classifiers contain very few neurons, they can still learn something while training on the specific probing task. So, by establishing a random baseline, there is a reference point to compare the evaluated linguistic tasks. If the performance for the random task and the linguistic task differ a lot, this would indicate a "high selectivity", which is favoured.

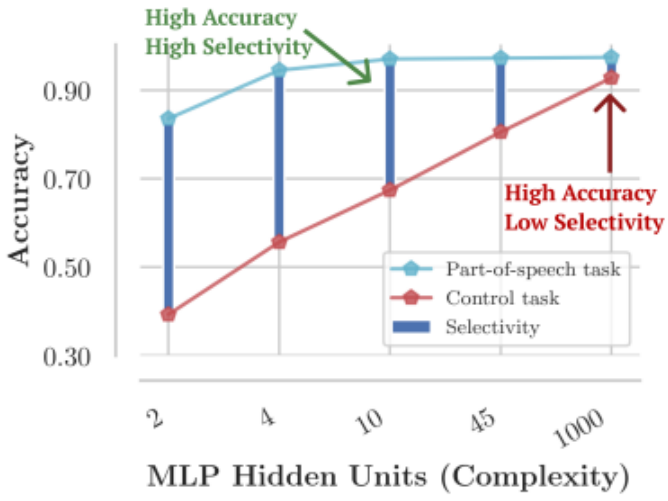

Figure 2: Selectivity is defined as the difference between linguistic task accuracy and control task accuracy, and can vary widely, as shown, across probes which achieve similar linguistic task accuracies. These results taken from § 3.5.

Figure 4 - Selectivity : Designing and Interpreting Probes with Control Tasks

If difference in performance between the linguistic task and control task is low this indicates the model achieved the same performance on the linguistic task as if it was randomly guessing. That would mean the model did not use its internal knowledge to solve the linguistic task but the knowledge was accidentally learned while training the probing classifier. So, for ensuring the model uses its internal knowledge. The study *Designing and Interpreting Probes with Control Tasks* (Hewitt & Liang, 2019) has found that popular probes on learned representations / embeddings from Language Model (ELMo) are not selective, suggesting that their high accuracy might be due to the probe's learning capacity rather than LLMs´ learned representations.

The study has also shown the effectiveness of regularization techniques like weight decay and reducing learned hidden state sizes are more effective than dropout layers in improving probe selectivity. Additionally, it has shown that layers of ELMo show varying levels of selectivity and accuracy in linguistic tasks, raising questions about which layer better encodes linguistic properties.

Permutation Task: This task assesses if the model learns at the sentence level or focuses on individual word elements by shuffling the order of words in the probed input sentences and asking the model to predict the label. This shows if the model needs to use the positional encoding representations of his encoder or if it succeeds without them.

2.5.2 *Prompting*

Prompting serves to activate the intrinsic knowledge within a language model by leveraging its pre-training. This method offers much flexibility since prompts can be designed in many ways. Especially for creative and open tasks this method is beneficial. However, this can lead to the following issues:

- Isolating specific linguistic capacities can require extensive prompt engineering for high accuracy (Liu et al., 2021, Section 7.2.2).
- In-context learning with prompt-based methods can be slow, limiting their utility when large datasets are involved (Liu et al., 2021, Section 7.2.2).

2.5.3 *Fine-Tuning*

Fine-tuning involves retraining a language model's encoder and potentially its decoder if present, on specific tasks or datasets to optimize its performance:

1. **Task-specific Retraining**: The LM undergoes further training on new data, changing the model’s parameters and encoded learning representations to better suit that task.
2. **Performance Optimization**: Fine-tuning aims to achieve maximum efficiency and accuracy for the LM on the selected dataset or task.

However, there is a risk that LLMs overfit or do not learn stably (Liu et al., 2021, Section 7.2.1).

2.6 Mann-Whitney U Test

Information in this section is summarized from Wikipedia´s Mann–Whitney U test (Wikipedia Contributors, 2019). The Mann-Whitney U test, named after Henry Mann and Donald Whitney, is a nonparametric statistical hypothesis test used for determining whether there is a difference in

the distributions of two independent samples. It is beneficial when the assumptions for a normal distribution cannot be satisfied.

- **Non-parametric Test**: Unlike t-tests that assume data follows a normal distribution, the Mann-Whitney U test does not require any assumptions regarding the data distribution. This makes it suitable for ordinal data or non-normally distributed interval data.
- **Independent Samples**: The test is applied to two samples that are independent of each other. In the context of this project those are usually referred as ‘none’, 'randomization' and 'permutation' data which represent different conditions being compared.
- **U Statistic**: The test ranks all the observations from both groups together and then calculates the sum of ranks for each group. The U statistic is essentially the number of times observations in one group precede observations in the other group in this ranked list.
- **P-Value**: This is the probability value that helps to determine the significance of the results. A p-value less than the chosen alpha level (commonly 0.05) indicates a statistically significant difference between the groups. In the Mann-Whitney U test, a lower p-value suggests that one group tends to have higher or lower ranks than the other, implying a difference in medians.
- **Two-Sided Test**: Indicates that the test checks for any significant difference between the groups, regardless of which one might have higher values.

3 Idea and Concepts

The focus of this Chapter is to outline the foundational ideas that inform the creation of linguistic tasks derived from discussion data and the subsequent decision to utilize these tasks for probing different language models. The aim is to scrutinize the intrinsic abilities that enable these models to understand and follow discussions.

3.1 Discussion Trees

Within the first week of thesis work, an exploratory phase unfolded, involving a study of Discourse Representation Structures (DRS) and Rhetorical Structure Theory (RST). While rich in linguistic insight, they were not immediately applicable to the task at hand. However, this foundational knowledge served as a crucial backdrop when my thesis main advisor, Andreas Waldis, introduced a parsed discussion tree dataset from Kialo. It is characterized by nodes representing claims and edges for relational stances between them. This discussion tree representation became the cornerstone from which the probing tasks would later be designed. It enabled the creation of tasks from a structure that visually articulated the complexity of debates. The nodes and their connections provided immediate insights and facilitated efficient translation into linguistic tasks without the tedious need for manual labelling, which is beyond the scope of this thesis. This approach allows for creativity to diverge from the more conventional/closed structures used in the field, such as the hierarchical and already well-explored frameworks of DRS and RST or more syntax-focused approaches. By harnessing the more flexible and less strictly defined properties of Kialo's discussion trees, a novel set of semantical probing tasks was envisioned.

3.2 First Impressions of Kialo.com

Kialo is an online debate platform that operates asynchronously. It is known for facilitating new and interdisciplinary topics and provides a fresh data source that has yet to be explored in the context of language model training. At this stage (December 2023), no evidence has been found to suggest that these models have been explicitly exposed to Kialo's content. This is therefore considered to be an unexplored area for investigation. Previous analyses in the field have focused on different data sets and applications, such as Reddit. However, the Kialo-based tasks represent a new dimension by integrating critical thinking and contemporary issues into the probing tasks. While contemplating various approaches to encapsulate the linguistic capabilities reflected in discussions, the overarching theme was to manifest this in a quantifiable, structured form aligned with the models' evaluation. The transformation from abstract constructs to precise applications was carried out in a recursive and reflective manner, generating a keen sense of the potential impact of the tasks.

At first glance, Kialo's debate platform reveals a systematic, tree-like architecture of argumentation, intuitively colour-coded to distinguish between affirmative and opposing points of view.

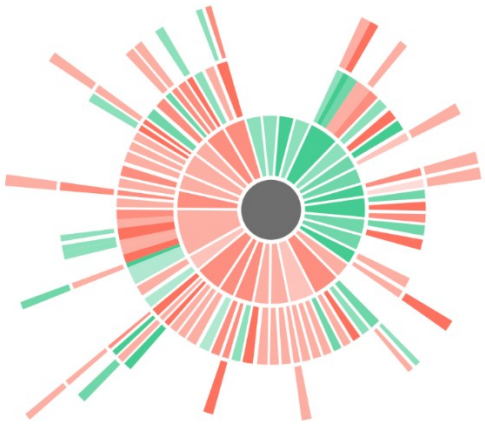

Figure 5 - Kialo.com discussion in sunburst style

This organisation, together with a broad, interdisciplinary selection of topics, paves the way for many participants to engage in probing tasks that test not just syntactic comprehension, but deeper semantic understanding - tasks that assess a model's ability to recognise the complex patterns of hierarchical thinking and argument dynamics inherent in Kialo's debates.

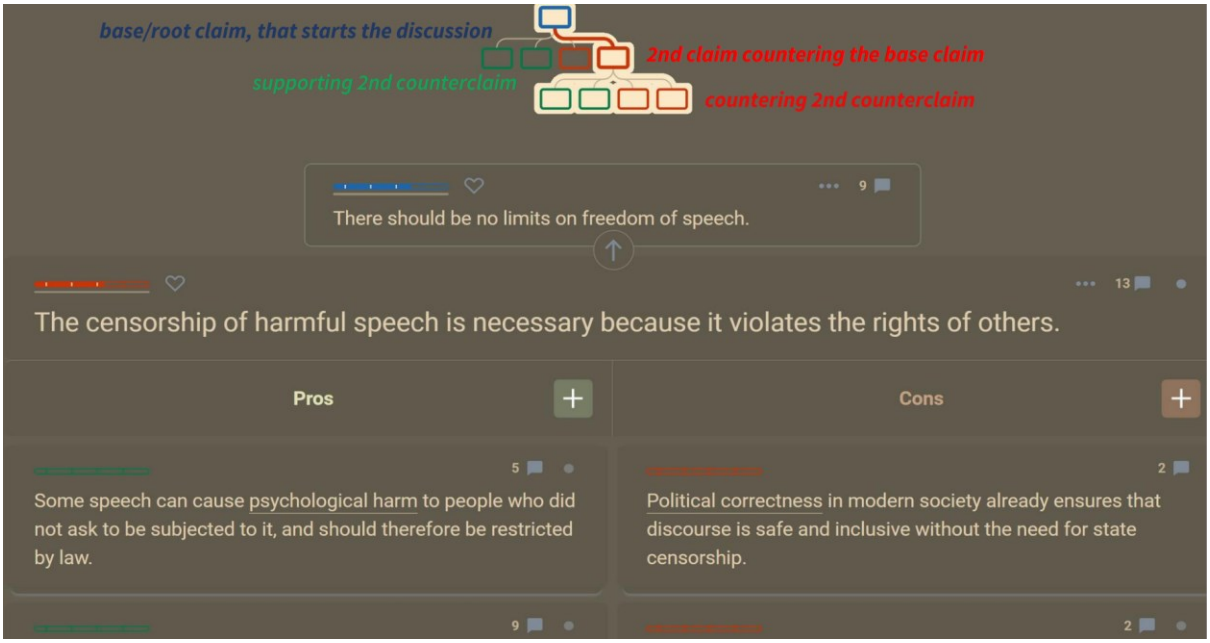

Figure 6 - Kialo.com discussion in tree style

Nevertheless, enthusiasm for Kialo's data must be tempered by an awareness of the biases inherent in the platform. Reliance on user content can lead to an unrepresentative range of viewpoints, potentially skewing a classifier's perceptions and compromising their ability to generalise across different styles of discourse. For probing tasks, the challenge is not in the structural detection of debate elements, but in ensuring that the classifier can accurately interpret the semantic essence of diverse and balanced arguments. Community biases, if present in the dataset, could, for example, affect the classifier's ability to judge the strength of arguments impartially, or to detect undercurrents of bias within the discussions themselves.

3.3 Probing Tasks Design Concept

The different discussion tree properties were identified iteratively, and the design concept was developed based on the discussion tree structure common to Kialo.com. Each concept reflects a different linguistic task derived from the tree properties, with the aim of covering a collection that is as diverse as possible and that takes into account different basic properties of human discussions.

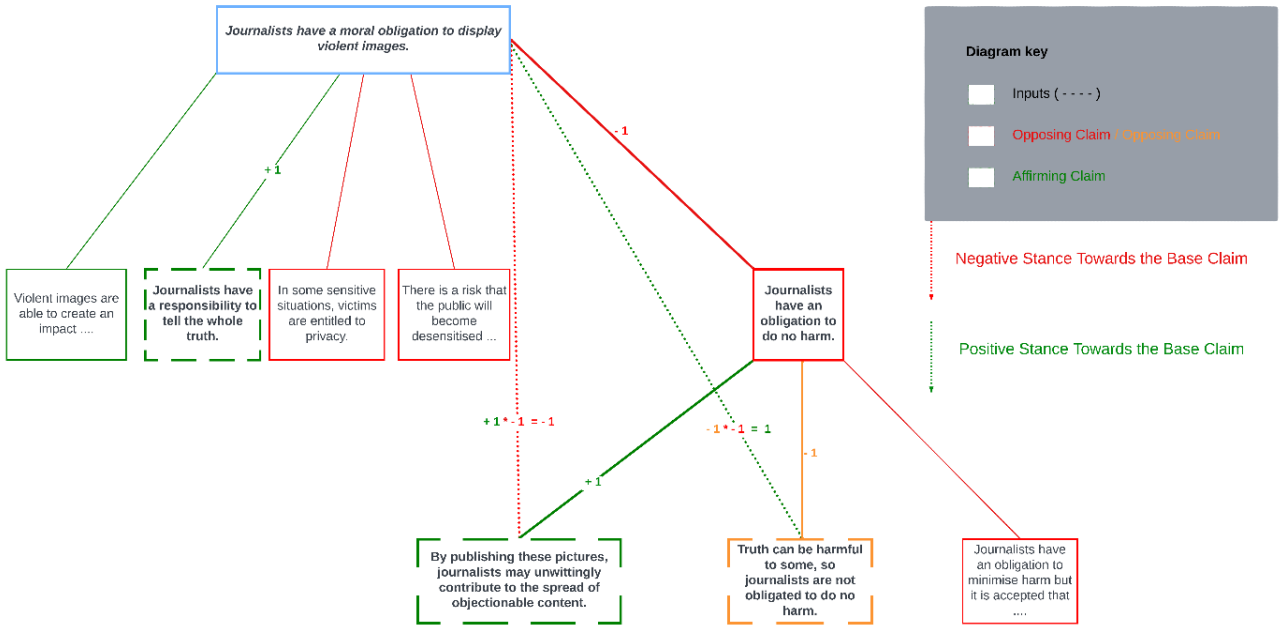

Figure 7 – Probing Design Task 1

The nodes and edges are the first linguistic properties that come to mind when studying discussion trees. So why not start with the edges themselves? They reveal the stances between different claims, and by following a claim path, recording the relationships, and finally multiplying them, long-range relationships can be inferred. This allows the creation of stance alignment examples with an adjustable distance between those claims, which sets the level of difficulty (whereas long-range context dependencies are often more complex to assess for LLMs). If we dare to compare this to an actual human trait of following discussions, it might

reflect how well we can understand someone's opinion or perspective, not just rationally within a context, but also emotionally.

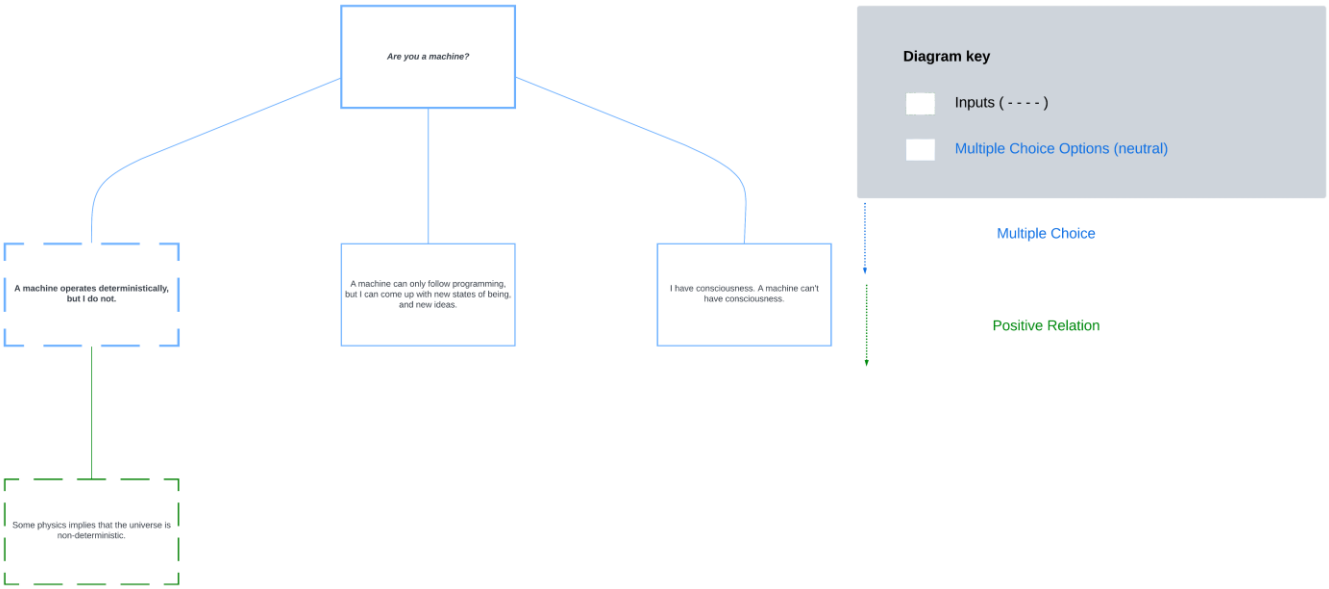

Figure 8 - Probing Design: Task 2

Another critical aspect of discussions is understanding their temporal relationships. The trees make it easy to see which claim follows which other claim. So this task is straightforward, generating labels based on the temporal relationships of claims encoded in the 'location' data frame already present in the first version of the dataset.

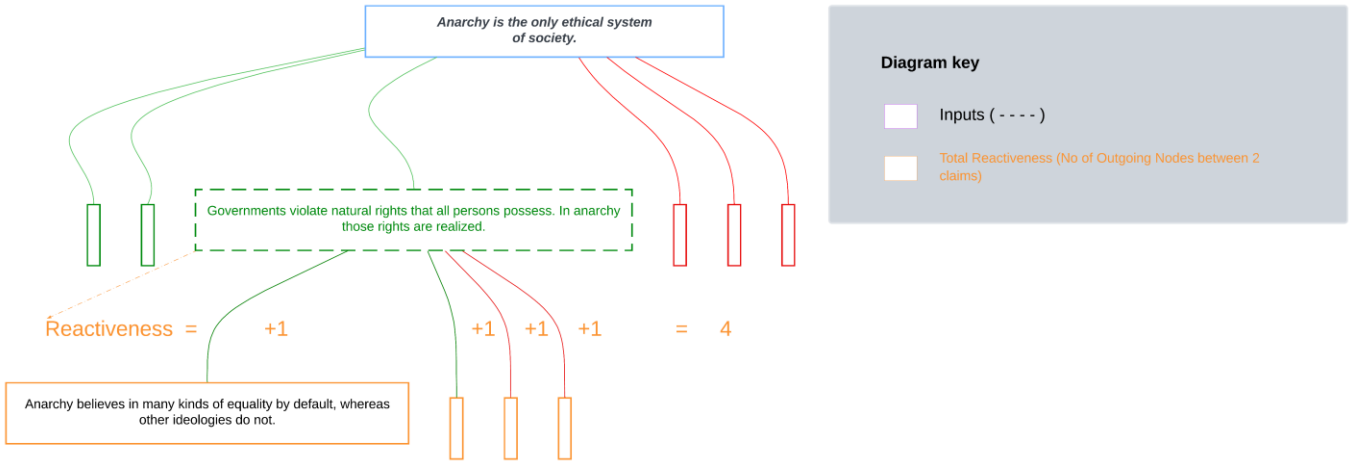

Figure 9 - Probing Design: Task 3

The outgoing/responding claims can be seen directly at the claim level. By simply counting these claims for each claim, another task is generated. This somehow corresponds to the level of reaction or level of provocation that a claim can trigger.

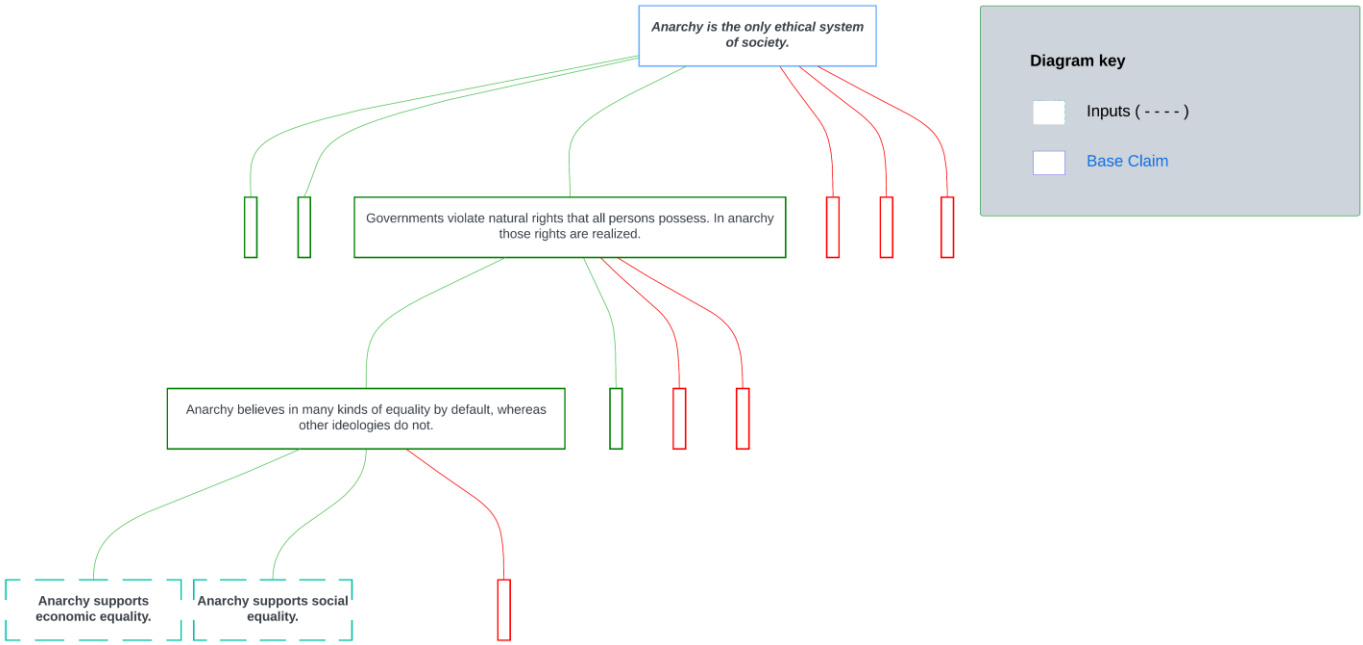

Figure 10 - Probing Design: Task 4

Another feature of discussion trees that is immediately obvious is the depth levels. Linguistically, this could reflect the degree of complexity of an argument in a discussion. It is always measured relative to the root claims (context), and by simply taking the shortest path from the root to any claim (which then always travels 1 level per 1 unit distance), the depth of each claim is quickly determined.

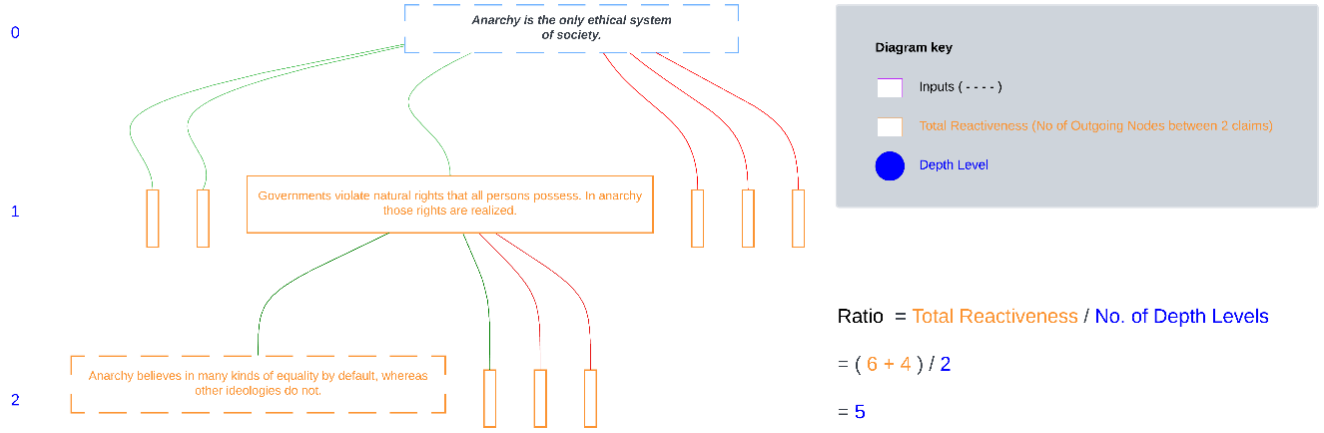

Figure 11 - Probing Design : Task 5

The last concept is a combination of the previous task and task 3: By taking the ratio of the depth levels travelled and adding the number of expected claims between a start and end claims, the result reflects the density or shape of the interspace: A number greater than 1 would indicate more breadth than depth, indicating a broader discussion with many possible reactions. A value less than 1 could indicate a more debated or focused discussion, where people often confirm or contradict their opinions. The model would be an attempt to predict this type of discussion, which, to be honest, would be a difficult task even for humans.

4 Methodology

This Chapter discusses the methodology used and how the answers to the research question should be achieved.

4.1 What Process Model is used in the Project?

Fundamental linguistic theories RST and DRS were studied first. The LLMs were selected immediately based on the main advisor's recommendation. The Kialo dataset and a script to parse discussions into a tree structure were also provided by my main advisor. Exploratory data analysis was carried out before designing probing tasks to identify tree features that could potentially be used for those. An iterative process guided the main development of the probing tasks.

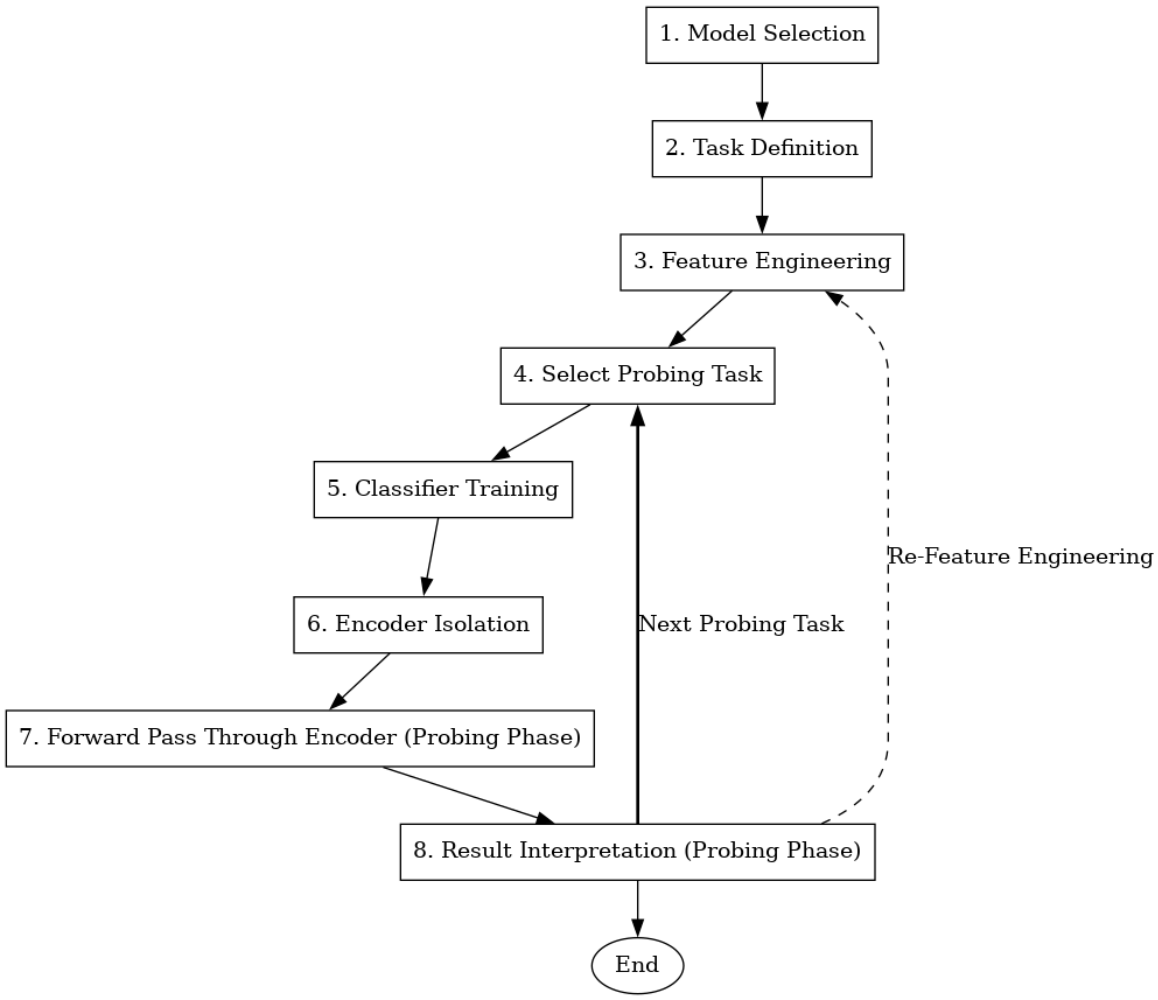

Figure 12 - Iterative Process Model

This approach assumed that an emerging understanding of the data would guide task generation through continuous inspection and refinement of the probing data. It also allowed for implementing simplified probing task versions by constraining the basic task version.

4.2 Which Properties are essential for following discussions?

Besides DRS and RST web research on discussions mainly originating from the field of psychology such as the *Cooperative Principle* (Grice, 1975) served as a starting point for understanding discussions properties. These properties were then recontextualized in light of neuroscientific findings to create a bridge to the quantifiable aspects of discussion trees.

4.3 What Implicit Understanding of Discussion can one expect from a LLM?

The selected LLMs and model groups' known capabilities and limitations were researched and summarized in Chapter 2 (e.g., *Hugging Face Summary of Selected LLMs*). The findings provided a baseline of expectations for LLMs' implicit understanding of discussions.

4.4 How to selectively evaluate LMMs on Tasks?

Probing was preferred to prompting and fine-tuning for evaluation because of its targeted and minimally invasive nature. Probing allows for a more selective assessment of the inherent capabilities of a language model without altering its parameters, as would be the case with fine-tuning. Prompting, while practical, often requires the complex creation of prompts, which can introduce confounding variables and is less effective in isolating specific language comprehension features. Moreover, prompting can become very slow as more data is used as described in Section 2.5.2. In addition, the introduction of a random baseline control task

provided a point of reference, enabling the calculation of selectivity (as detailed in 2.5.1) as a metric for comparison.

4.5 How can Properties be verified within LLMs using Probing Tasks?

Probing tasks were structured using authentic, multidisciplinary discussion data from Kialo.com to validate essential communication properties within language models. The discussions were parsed into a graph format, with nodes depicting individual claims and edges illustrating their supporting or opposing connections. This transformation enabled the application of standard tree operations, such as calculating the maximum depth through the longest path in a discussion, enhancing the grasp of how online dialogues correlate with tree-based structural properties. These operations facilitated the automated creation of labels by using tree properties.

Moreover, probing tasks were developed to correspond to the identified essential communication properties grounded in (neuro)psychology within the constraints of the tree data structure. Probing provided a focused measure of the probing task while avoiding interactions caused by confounding variables. By evaluating LLMs from different architecture groups, further biases caused by these differences and training data can be reduced but not entirely excluded. The findings on the known limitations and capabilities of the LLMs have also placed further constraints on the research question.

4.6 How do various LLMs differ on the Probing Tasks and which Properties are crucial?

To answer this question several sub questions were defined.

4.6.1 How can LLMs Performance be compared?

LLMs were subjected to a series of probing tasks under uniform metrics and controlled conditions, including variation across different folds and seeds. The performance metrics for each LLM on individual tasks were consolidated by calculating median F1 scores to ensure a standardised basis for comparison.

The analysis continued with the calculation of normalised differences, contrasting the median F1 scores from the 'NONE' condition (standard condition), with those obtained on the 'NO CONTEXT' and control tasks (“RANDOMISATION” and “PERMUTATION”).

LLMs results were interpreted with respect to their known capabilities/limitations (training data, architecture) as collected and summarized in Chapter 2.

4.6.2 How can LLMs Robustness be compared?

This project utilized Analysis of Variance (ANOVA) to evaluate the robustness of various LLMs to changes in seeds and data folds. ANOVA is a statistical method used to compare the means of multiple groups to determine if there are significant differences between them. In this context, the groups are defined by different seeds and folds used in the probing framework on Juypterlab. The choice of ANOVA was driven by the need to understand the stability and consistency of LLMs under varying initial conditions. Specifically, the analysis aimed to ascertain how the models' performance fluctuates with changes in:

- **Seeds**: Seed variations can affect the initialization of model weights and training data shuffling, potentially leading to differences in model learning and performance.
- **Folds**: Variations in data folds influence the specific training, development, and evaluation datasets used (e.g., for training the probing classifier), which can impact the generalization ability of the models.

For assessing seed robustness the fold variable was held constant and for assessing folds variance the seed was held constant.

The ANOVA results, particularly the p-values, indicated the degree of influence that seed and fold variations had on each model. Lower p-values signify that changes in seeds or folds lead to significant variations in performance, suggesting lower robustness.

4.7 How can Bias be mitigated when evaluating LLMs to obtain meaningful Results?

To address the complexity of mitigating bias in LLMs through probing several measurements were taken:

- **K-Fold Organization**: Four-fold cross-validation was employed where each probing example from the Kialo dataset was allocated once to the test set, once to the development set, and twice to the training set. This specific distribution across the folds, exemplified by Kialo's claims and topics, aimed to negate any bias from the folds.
- **Seed Variation**: Five different seed numbers were used for each probing task to counter initial condition biases. Each example was cycled through as a test, development, and twice into the training set to affirm that sensitivity to seed variability did not skew results.
- **Kialo Dataset Characteristics**: Considering the inherent randomness present in the Kialo platform—where various users initiate discussions and themes asynchronously—the probing data leveraged this aspect to enhance the natural diversification of examples.
- **Feature Selection**: A careful selection of features was performed before converting the dataset into discussion trees. Factors manifesting strong correlations or deemed outside the thesis scope were removed. This was informed by relevance to the thesis aims and pragmatic constraints, such as time limitations.

- **Probing data cleaning**: This step involved the removal of irrelevant features from the exploratory inputs, such as hyperlinks, which could potentially bias the model towards non-essential patterns, thus preventing the models from being confused by irrelevant connections.
- **Architecture diversification**: Models were selected from different architectural lineages, such as BERT, GPT, and ELECTRA, ensuring that the scope of the evaluation is not limited to a particular type of learning or representation bias.
- **Robust Metric**: The F1 metric was chosen for its robustness, providing a balanced evaluation of precision and recall. This choice was particularly pertinent given the engineered balance in label distribution and uncertainties that may arise from the dataset's complexity or scope.

4.7.1 How is the Reproducibility of Results ensured?

Random seeds were set to recreate random sampling while generating the probes and for recreating initialization while running the probing tasks on Juypterlab.

4.8 What Libraries and Frameworks are used in the Project?

This section shows an overview of the tools used in this project. The data analysis and the engineering of the probing tasks were done with the known Python libraries:

4.8.1 Which Libraries are used for Cleaning, Analysing and Selecting the Data?

- **pandas**: Standard table operations and calculations: data selection
- **matplotlib.pyplot**: Data visualisations such as boxplots, bar plots, curve diagrams, histograms for count distributions, boxplots to show outliers
- **seaborn:** Several data visualisations

- **plotly.graph_objects :** Visualises interactive 3D plots
- **scipy.stats:** Calculates z-score for outlier analysis, Shapiro-Wilk Test for verifying / falsifying normal distributions (yes if p-value > 0.05)
- **igraph:** Parses discussion tree structures
- **re:** Data selection by regular expressions
- **tqdm:** Outputs progresses such as data generations for the probing tasks
- **langdetect:** Detecs English claims

4.8.2 Which Libraries are used for handling different File Types?

- **pickle:** Loads pickle files. Pickle is an efficient way of saving data
- **yaml:** Reads YAML formatted files required for the configuration of probe tasks into the probe framework

4.8.3 What other Libraries are used?

- **ast.literal_eval**: Verifies/Falsifies the format of inputs before running the probing tasks on Jupyterlab
- **gc:** Garbage collection to free RAM memory
- **random:** Random operations such as setting seeds and sample data

4.8.4 Which IDEA / Servers are used?

- **Google Colab & Dataspell** for data analysis and probing task engineering because Colab proved itself to be faster for some data operations especially while using igraph objects
- **Juypterlab**: Connected to HSLU GPUs to run the LLMs on the engineered probing tasks

4.9 How are Discussion Graphs analysed?

Various methods were employed to dissect the architecture of online dialogues. The depth and breadth of discussion trees were measured, and centrality measures were applied to identify trends and common patterns.

4.9.1 Depth and Breadth

These properties were calculated using the following formulas:

Discussion Depth:

What It Measures: The depth represents the number of claims or arguments along the longest path in a discussion thread.

Unit of Measure: Denoted as D where D belongs to the set of non-negative integers {0, 1, 2, ..}.

$$\text{Depth} = \max(\text{Number of edges in the longest path from base claim to a target claim})$$

Discussion Breadth:

What It Measures: The breadth represents the average number of direct unique responses to a claim at each level of the discussion.

Unit of Measure: Denoted as B where B belongs to the set of non-negative float numbers{0.01, 0.02, …) at a specific depth d.

$$\text{Breadth} = \frac{\text{Total number of child nodes}}{\text{Total number of parent nodes}}$$

4.9.2 *Centrality Measurses*

For analysing and expecting the structure of discussion following centrality measurements based on graph theory were used and interpreted according to the given definitions , where N is the total number of nodes in a discussion network/graph.

Degree Centrality (C_D)

Reflects the number of direct connections a claim has within the graph.

$$C_D(v) = \frac{\deg(v)}{N-1}$$

where

- $deg(v)\ is\ the\ degree\ of\ the\ vertex\ v$
- $C_D(v)\ ranges\ from\ 0\ to\ 1$
- $N\ is\ total\ number\ of\ nodes\ in\ the\ graph$

$For\ C_D > \frac{2}{N}: Well\ connected$

$For\ C_D < \frac{N}{4}: Lesser\ connectivity$

Closeness Centrality (C_C)

Indicates how close a claim is to all other claims in the graph.

$$C_C(v) = \frac{1}{\sum_{u \neq v} d(u,v)}$$

where

- $d(u,v)\ represents\ the\ shortest\ path\ distance\ between\ vertices\ u\ and\ v.$
- $C_C(v)\ ranges\ from\ 0\ to\ 1$
- $N\ is\ total\ number\ of\ nodes\ in\ the\ graph$

$For\ C_C > 0.6: More\ central$

$For\ C_C < 0.4: Peripheral$

Betweenness Centrality (C_B)

Measures the extent to which a claim lies on the paths between other claims.

$$C_B(v) = \frac{\sum_{s,t \neq v} \sigma_{st}(v)}{\sigma_{st}}$$

$where$

- $\sigma_{st}\ is\ the\ total\ number\ of\ shortest\ paths\ from\ vertex\ s\ to\ vertex\ t.$
- $\sigma_{st}(v) is\ the\ number\ of\ those\ paths\ that\ passt\ hrough\ vertex\ v.$
- $C_B(v)\ ranges\ from\ 0\ to\ \frac{1}{2}(N-1)(N-2)$

$For\ C_B > \frac{N^2}{4}: Key$ bridges

$For\ C_B < \frac{N^2}{8}: Less\ central$

4.9.3 Community Connectivity

Has been used to predict how fragmented the discussion graphs are:

$$\text{Connectivity} = \frac{\text{Inter-community edges}}{\text{Total number of communities}}$$

4.9.4 Outlier Detection

For outlier detection, the z-score with a threshold of 3 was used.

5 Implementation

This central Chapter is about the technical implementation (data analysis, probing task engineering, probing the LLMs on those tasks).

5.1 Defining essential Properties based on Research

The information of this section is based on Wikipedia research on Grice's *Cooperative principle* (Wikipedia Contributors, 2019b). The first part of this small research focused primarily on psychology. Specifically, Grice's cooperation theory helps identify some fundamental capabilities needed for understanding discussions: maxims, quantity, quality, manner, and relation. One should be aware that those principles vary across cultures, are descriptive, and are not considered strict rules. So, for the first version, the following properties were defined:

1. **Effective Quantity Management**: The capability to adjust the communication amount to the given context and time frame directly influences the understandability and follow-through of a discussion.
2. **Knowledge Awareness and Accuracy**: The ability to discern relevant and accurate information based on semantic properties.
3. **Appropriateness**: The capability to use appropriate verbal and non-verbal communication techniques.
4. **Contextual Understanding and Relevance**: The grasp of words in the context of varying discussions, including the relational structure and the thematic relevance. In a broader sense, it also grasps the discussion types. For example, how the discourse unfolds within a debate is markedly different from that of a casual conversation; the former may be structured around articulated stances and counterarguments, whereas the latter might

flow more freely with fewer explicit constraints on argumentative structure and response patterns.

5. **Self-Evaluation and Adjustment**: Continuously assessing and adapting one's communication approach.

In the second step, the properties are redefined, integrating also anatomical entities discovered by neuroscience.

1. **Effective Quantity Management:** The capability to adjust communication output in response to the demands of incoming information, analogous to working memory span.
2. **Context Processing:** Neuroscientifically, it aligns with how the brain comprehends and navigates complex environments, utilizing areas involved in spatial awareness, executive functions, and memory. The prefrontal cortex, known for its role in complex cognitive behaviour and decision-making, is pivotal for this task (Tran et al., 2021). This refers to the brain's ability to process and integrate its contextual elements in discussions. More specifically, for discussion trees that means understanding the structure and hierarchy of the conversation – how different claims or comments are connected, the flow from one point to another, and the relationship between various parts of the discussion (e.g., how a response relates to a previous statement).
3. **Theory of Mind Encoding:** This represents the ability to understand and take into account another individual's mental state (Premack & Woodruff, 1978). The biological components responsible for this are the mirror neurons. (Rajmohan & Mohandas, 2007)
4. **Emotional and Social Cognition:** The understanding and navigation of emotional components within conversations, grounded in the amygdala's role in affective processing (Adolphs, 2010).

5. **Self-Evaluation and Adjustment**: Continuously assessing and adapting one's communication approach.

These definitions, of course, are only partially representative for assessing discussion understanding precisely, given the fact that we are far from having discovered the entirety of the human brain. Nonetheless, they serve as a reference point grounded on research until 2023.

5.2 Aligning researched Properties and Discussion Trees

Next, the defined neuroscientific and psychological properties are aligned with concrete features of the discussion tree conceptualised in Section 3.3.

1. **Stance Alignment** probes the LLM's ability to identify individual stances in a discussion, effectively operationalising the psychological aspect of Theory of Mind. This task assesses the LLM's ability to simulate the functionality of human mirror neurons, which facilitate the understanding of another's mental state.
2. **Sequential Coherence** probes the LLM's ability to logically link sequences of dialogue, mirroring the role of the prefrontal cortex in human context processing and decision making. It measures the potential of LLMs to replicate the cognitive processes underlying human temporal awareness in conversations.
3. **Interactive Dynamics** probes how effectively the LLM can anticipate interaction patterns, in particular the number of responses a claim is likely to elicit, reflecting the human trait of appropriateness and cognitive load management.
4. **Claim Depth Hierarchy** probes an LLM's understanding of the layered structure of discussions, reflecting the human cognitive process associated with contextual

understanding and relevance - a skill that involves grasping the interrelationships and thematic significance within a conversation.

5. **Discourse Contour Recognition** probes how well the LLM predicts the structure and connectivity of a discourse, synthesising features related to effective quantity management and context processing.

These probing tasks are intended to mirror known human capabilities used for understanding complex discussions, drawing a parallel between research-supported cognitive properties and machine learning capabilities. Given the fact that the human brain is still far from being understood they are not a substitute for human cognitive processes, but a reflection of the potential that LLMs have when matched to a selected set of human discussion comprehension. The probing methods and the resulting analysis are therefore designed to add a dash of empirical substance to theoretical investigations of LLMs' abilities to follow human-like discussion patterns.

5.3 Exploratory Data Analysis for unparsed data

In order to get a better sense of the underlying discussion data structures and exact features, the initial web-scraped Kialo dataset provided by my main advisor is analysed first.

5.3.1 Data Structure

The dataset shows a list of JSON structured discussions in a pickle file format accompanied by the following (here slightly adjusted) descriptions:

The **`dumped_discussions_first.pickle`** includes a 17.832 discussions.

Structure of discussions :{

"title": Title of the Kialo discussion

"background info": Background information of the Kialo discussion

"discussion": {

"claims": [...] All claims of the full discussion, more details later

"locations": [...] Location of the claims, more details later

"touchedClaimOrLocationIds": not relevant }

}

Structure of claims :{

"id": The id of the claim itself in format XXXX.YY where XXXX is the discussion id, when YY is equal to 0 it is the root claim

"authorId":"0077c272-1fdc-4202-a06a-160482dfc12d", The id of the author

"created": 1382962412430, Unix Timestamp of the creation date

"version": 2, Number of times the claim has been edited

"text": "Anonymous currency discussions", the text of the claim, often empty or not relevant for root claim (i.e. 333.0)

"lastModifiedForSitemaps": 1382962412430 }

Structure of positions

There is a location entry for every claim that is not the root. This entry describes the position of the claim and, therefore, its target.

{

"id":"333.1228", id of the relation

"targetId":"333.355", id of the source claim

"version":1, Number of times the relation has been edited,

"isOrigin": true,

"authorIdentityId":"54d22c9853057c01262a7192",

"created":1395770583629,

"parentId":"333.338", id of the target claim

"relation":1, value is 1 if relation is Pro, value is -1 if relation is Con

"isDeleted": false, ignore relation if it is true

"**lastModifiedForSitemaps"**:1520166022299

}

Observations:

- The same node (see 333.1) is being saved n times, whereas n = number of outgoing edges
- Some IDs are skipped/missing/deleted.
- Nodes and edges being saved in igraph formatted sequences.
- Generally the position ID consists of two parts: the ID of the discussion before the comma and after the comma follows the ID of the respective claim within this discussion. Although incremental enumeration of claims is often observed, a higher claim ID is not necessarily written later.: Sometimes claims are deleted again and continuation starts

from the deleted positions and from the temporal sequence (next free id is taken). To have the precise order the targetIds and parentIds must be taken (which indicate to which claim (id) a claim responds and which claim that claim follows.)

5.3.2 Distributions

To use familiar Python methods (e.g., pandas) for analysing the data, the initial dataset is converted into three data frames: the discussion_df, the position_df, and the claim_df. The analysis starts with generating box plots for count distributions and outlier histograms. This gives a sense of the diversity of the data used:

Claims Per Author

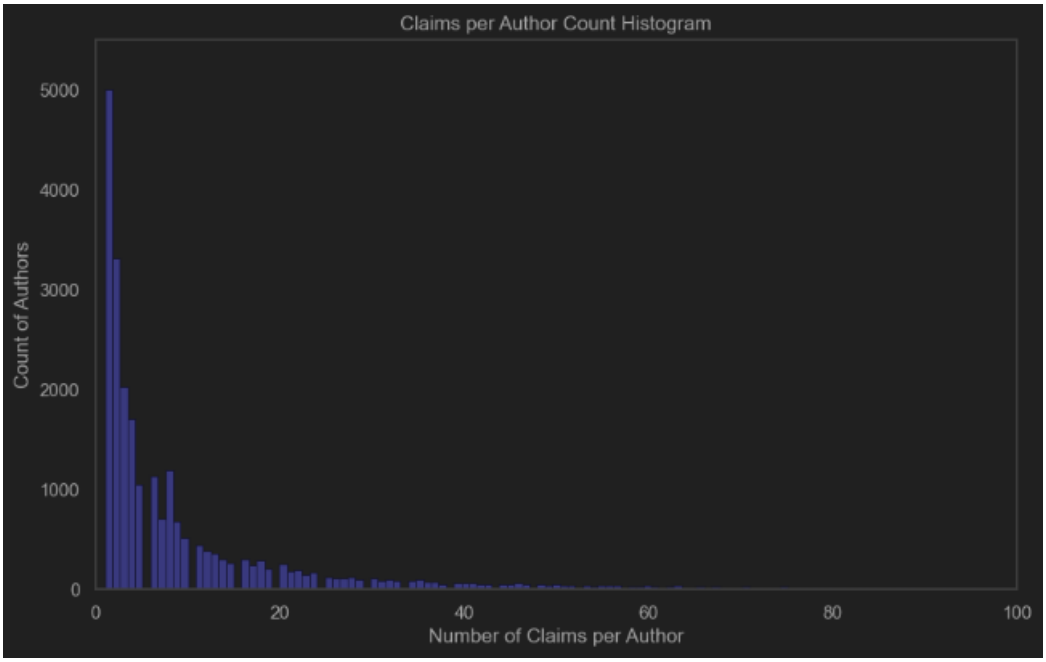

Figure 13 - Claim per Author Count Histogram

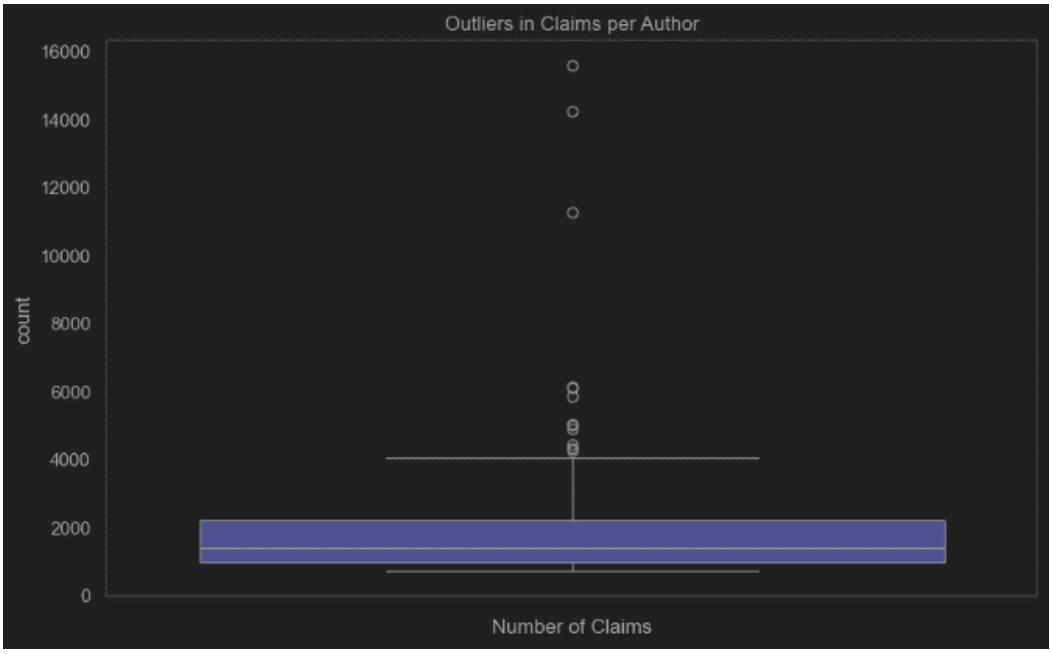

Figure 14 - Claims per Author : Outlier Boxplot

- The Kialo dataset comprises 746,630 claims made by 24,736 unique authors.
- The median number of claims per author is 5, indicating that most authors contribute a handful of claims.
- 170 outlier authors who contribute a significantly higher number of claims, suggesting the presence of very active participants within the platform.

Unique Topics per Author

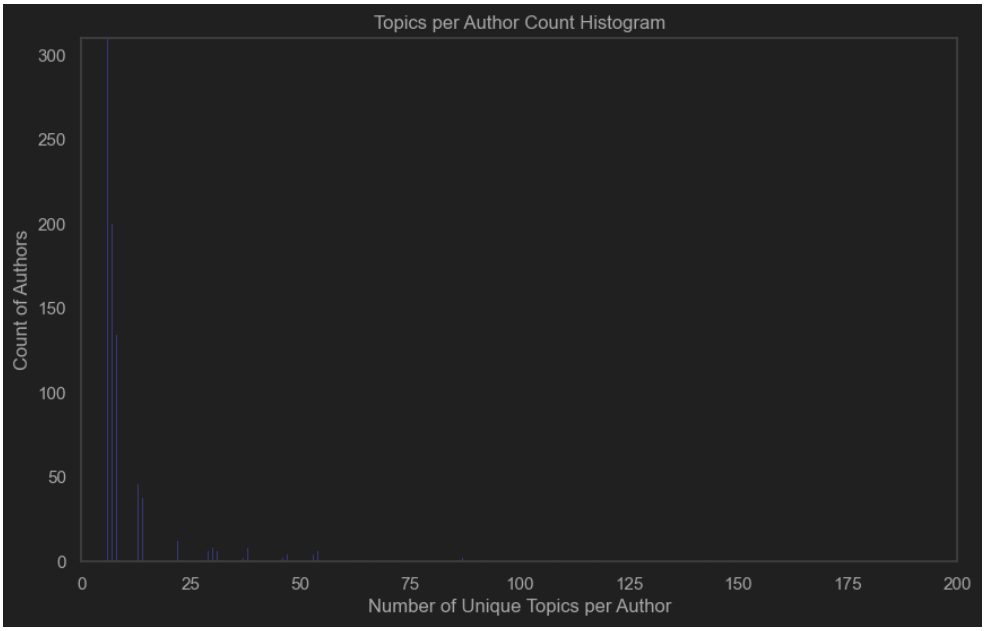

Figure 15 - Topics per Author Histogram

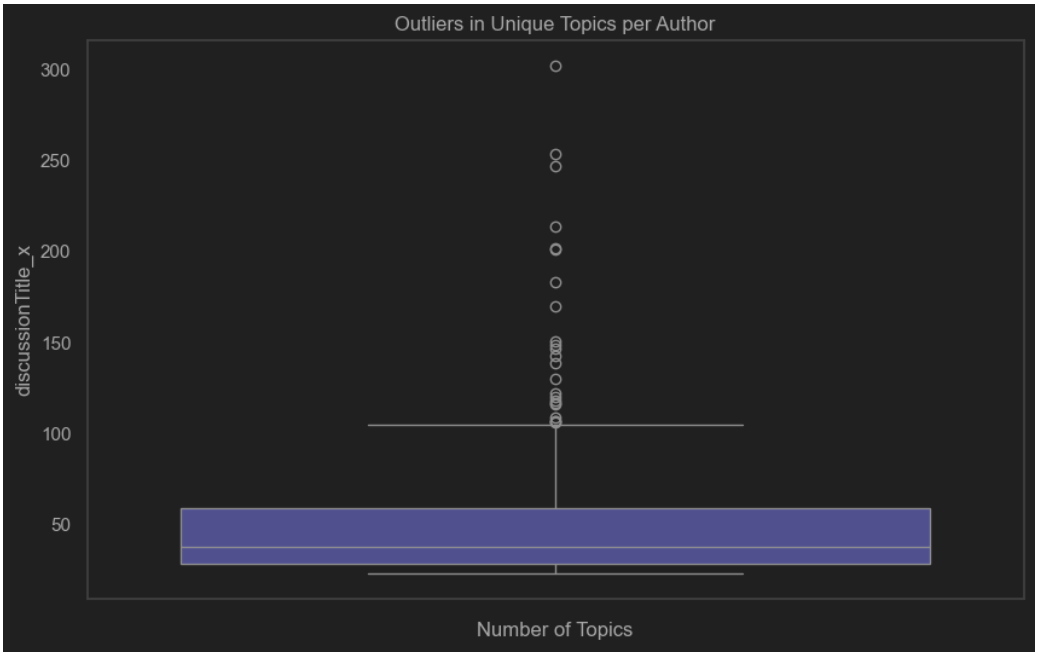

Figure 16 - Topics per Author Boxplot

- Each author typically discusses one unique topic, with the median at one unique topic per author.
- 238 outliers who discuss a more comprehensive range of topics, showing that a minority of authors have broader engagement across the platform.

Claims Per Topic

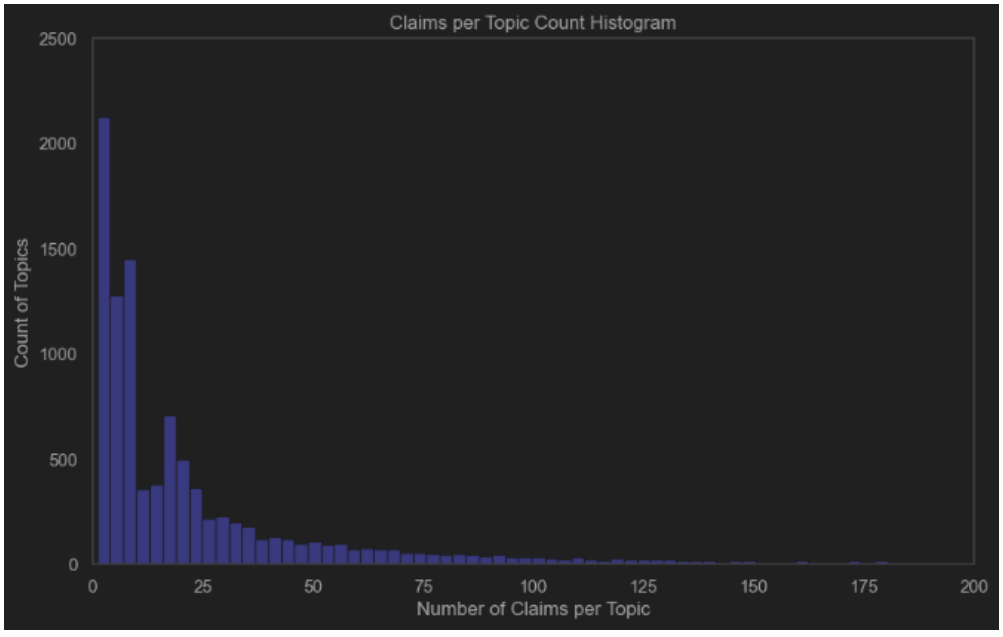

Figure 17 - Claims per Topic Histogram

Figure 18 - Claims per Topic Boxplot

- There are 10,758 unique topics with a median of 14 claims per topic.
- 114 outlier topics have attracted a disproportionately high number of claims, possibly reflecting hotly debated issues or areas of solid community interest.
- The largest topics are analysed at a later point in time but fit here well in representing the outlier topics with the greatest number of claims.

Top 40 Discussions by Number of Claims

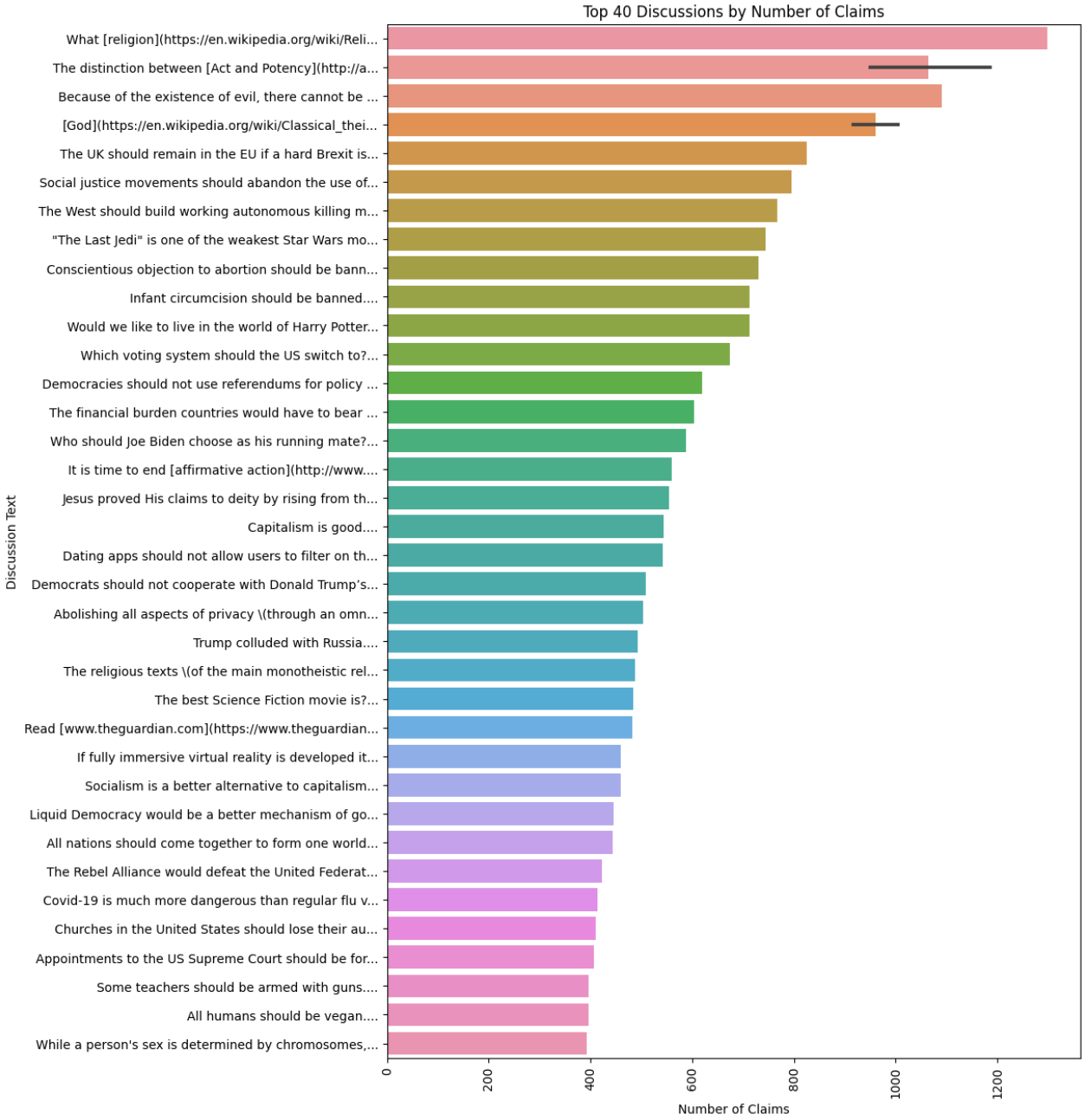

Figure 19 - 40 Largest Discussions

The following table summarizes the analysed distributions:

Entity	Total Number	Median	Unique Number	Number of Outliers
Claims per Author	-	5.0	576.0	170.0
Claims per Topic	-	14.0	650.0	114.0
Topics per Author	-	1.0	106.0	238.0
Claims	746,630	-	225716.0	-
Topics	-	-	10758	-
Authors	-	-	24736	-

Stance Distribution (Pro /Con Relations)

This shows the overall count distribution of relations which are found as edge attributes in the positions dataframe across all discussions:

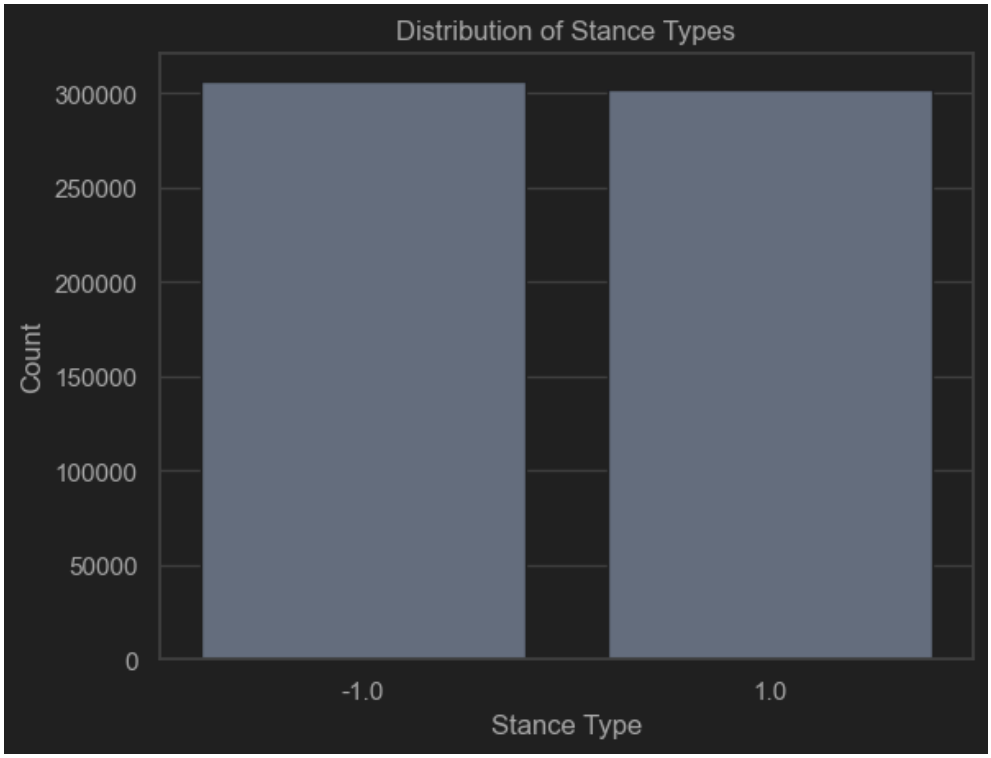

Figure 20 - Stance Distribution

- The distribution of pro and con stances within the claims is not evenly split, with 306,343 instances of con (against) positions and 301,980 of pro (supporting) positions, excluding neutral stances. This close number between pro and con stances indicates a balanced representation of differing viewpoints on the platform.

Key Conclusions

- Active Minority: A small group of authors contributes a large number of claims, indicating the presence of highly active individuals.
- Focused Engagement: Most authors tend to focus their contributions on specific topics rather than spreading across multiple subjects.
- Debate Hotspots: Certain topics garner significantly more claims, which may point to them being contentious or of high interest to the Kialo community.
- Balanced Debate: The nearly even split between pro and con stances suggests a healthy balance of opinions on the platform.

Topic Distribution

Since no labeled topics are in the Kialo dataset, those must be generated differently. By asking a BERT LLM (masked token) to predict the category of each theme (most likely) out of the base claim text, the following distribution is obtained:

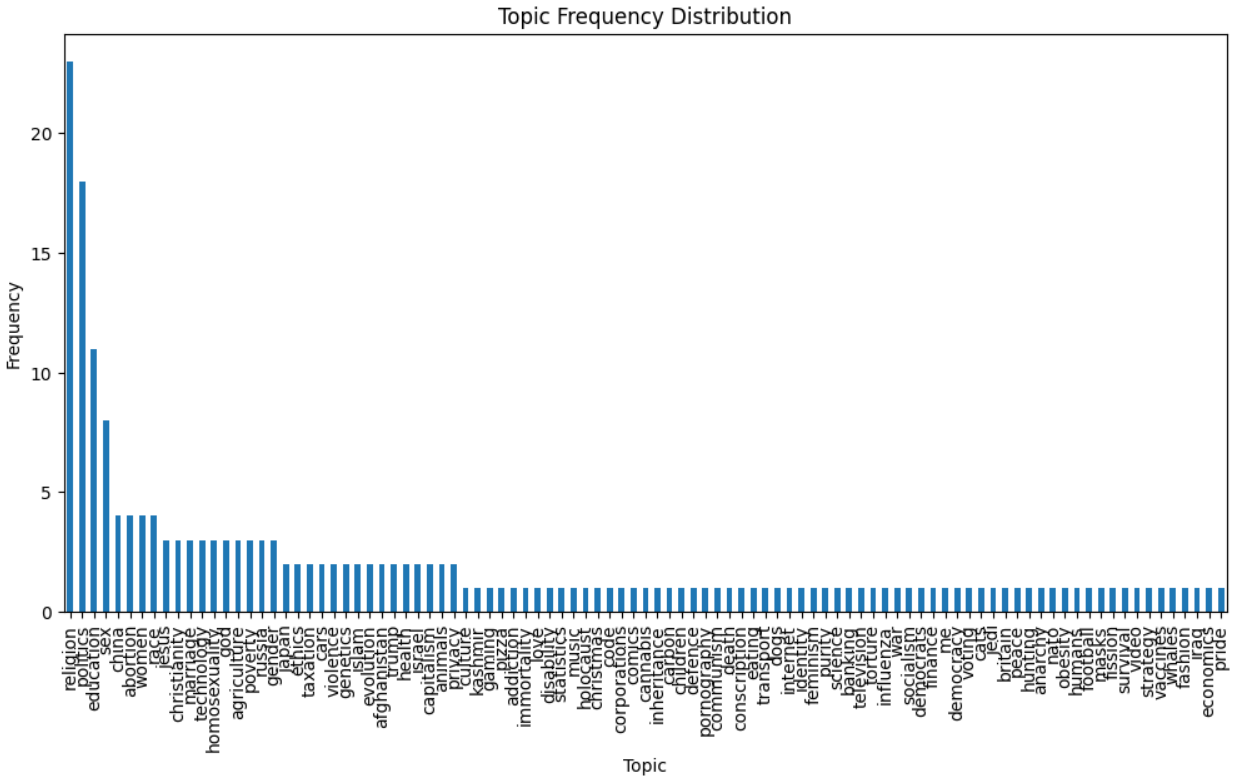

Figure 21 - Topic Frequency Distribution

5.3.3 Feature Selection

To identify redundant features a correlation heatmap is plotted:

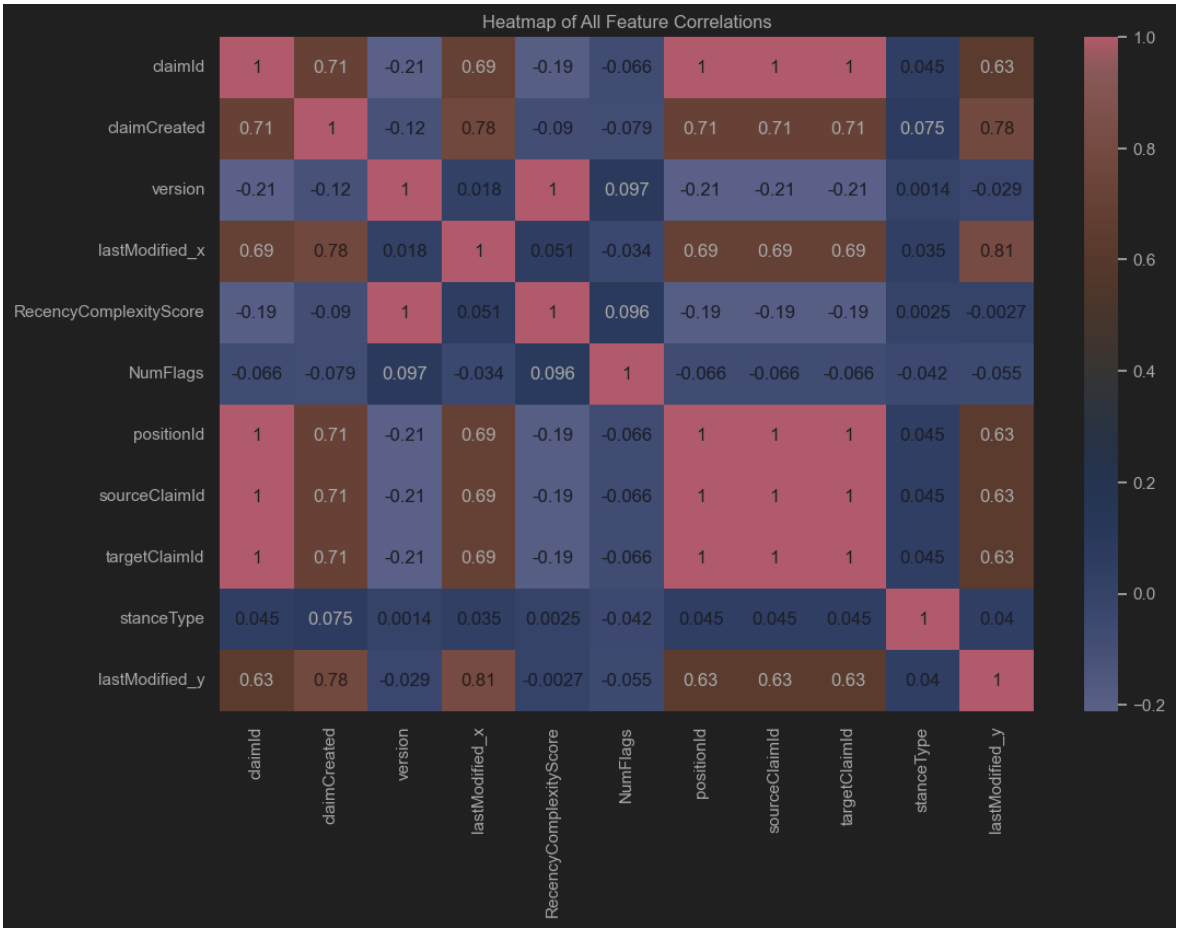

Figure 22 Heatmap of Feature Correlations

Most of the features are uncorrelated. There is a stronger correlation between timestamps and IDs or in-between IDs but those are negligible.

The following features were removed since they were not present in sufficient amounts or were not relevant to the scope of this project:

- **`deleted positions`** Used to filter out deleted claims when parsing the data with the parse_discussion_graph function (described in the next section) and then removed.

- **`flags`** are removed: Useful feature to assess the quality of a claim though most claims have 0 flags: 22721flags are found in 679485 claims: so flags exist for only 3% of all claims
- **`copierId` , `accepterId` and `discussionLinkTo`** of positions and claims data frames have been removed because they do not provide any useful linguistic information.
- Time-related features

  Since the sequential information is captured in the df_positions data frame using sourceId (origin claim) and targetId (destination claim) the following time-related features are removed:

  - **versions** : Count: 679,485, Mean: ~1.8 ,Max: 46

    Most claims have a version number around 1-2, indicating that many claims are not modified multiple times.
  - **Claim Created Timestamps**: Broad distribution, possibly related to the age of the claim.
  - **lastModified Timestamps:** There is a moderate correlation between 'ClaimCreated' and 'LastModified', confirming that the date, when the claim was created is mostly the date when it was last modified. Therefore, most claims have only 1 version: the 1st and last.

No significant correlations found between stanceType and IDs (only correlations between Ids.) which ensures the stanceType (positive or negative relation for a claim to another claim) shows no pattern / regularity throughout.

5.4 Discussion Graph Parsing

My main advisor provided a script for parsing the filtered discussion data such that relationships between the traditional tree features and potentially usable linguistic features for probes are more evident. So, the “parse_discussion_graph (discussion)” function is applied to the JSON-framed initial dataset. All discussions are represented as one big graph, while a subgraph signifies a specific discussion. Claims are represented as nodes and relations as edges.

The following images show a subdiscussion in different node representations:

Raw Subgraph

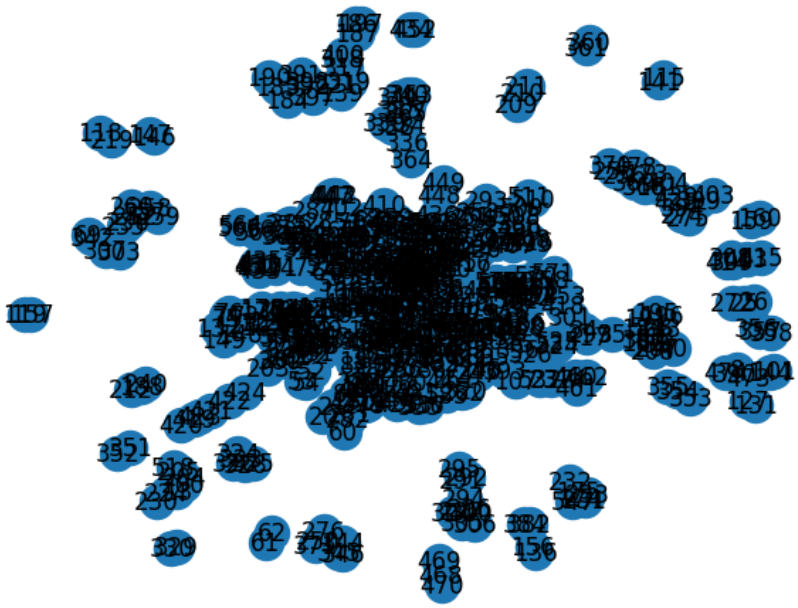

Figure 23 - Raw Nodes Representation

Subgraph (Kamada-Kaway)

The Kamada-Kaway algorithm visually represents the raw subgraph nodes, which helps to see specific patterns (e.g., claim count). It is not meant to represent the exact structures.

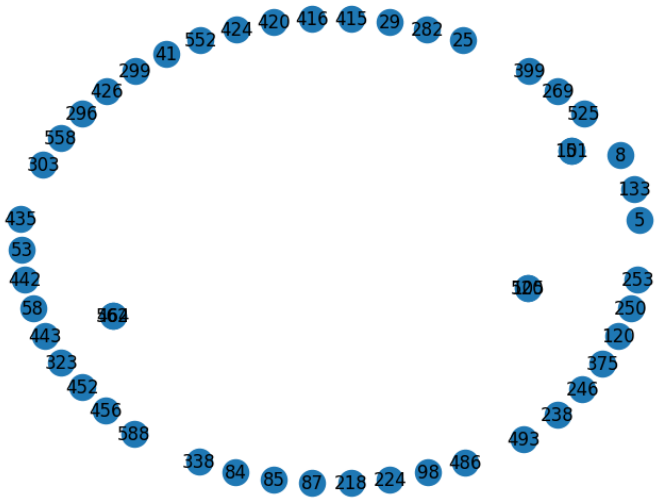

Figure 24 Kamada-Kaway Subgraph Representation

Nodes and Relations (Pro/Con)

The following shows the root claim (topic) in the center to which multiple claims respond. The stance of the responses in relation to the root claim (topic) are colored green for affirmative claims and red for opposing claims.

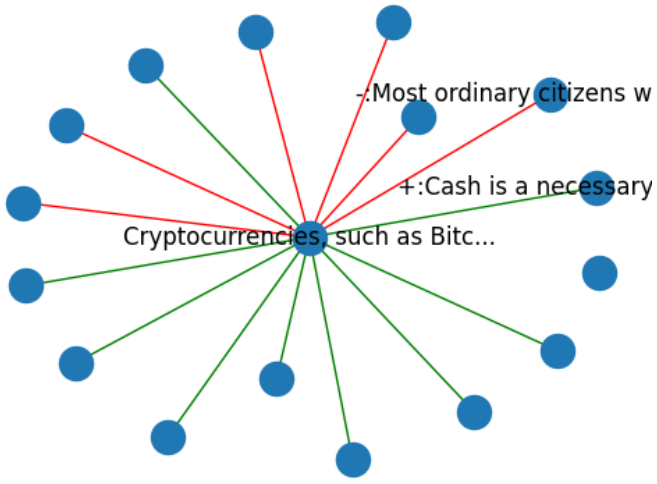

Figure 25 - Nodes & Relations Subgraph Representation

Discussion claims distribution in 3D

The 3D visualization of a subgraph shows the overall shape and how single claims are distributed irregularly throughout depths and breadths.

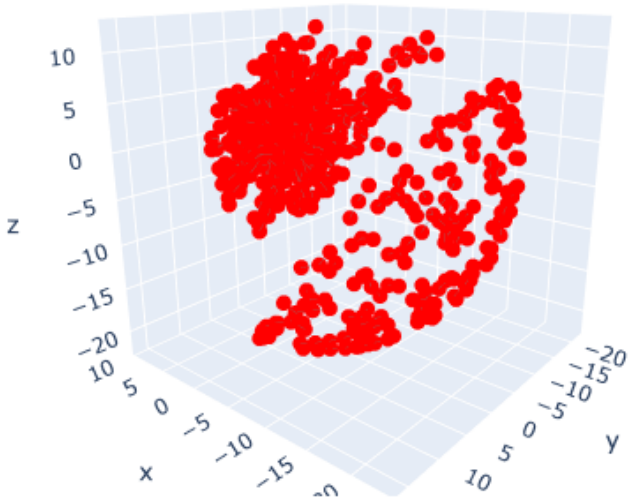

Figure 26 - 3D Subgraph Representation

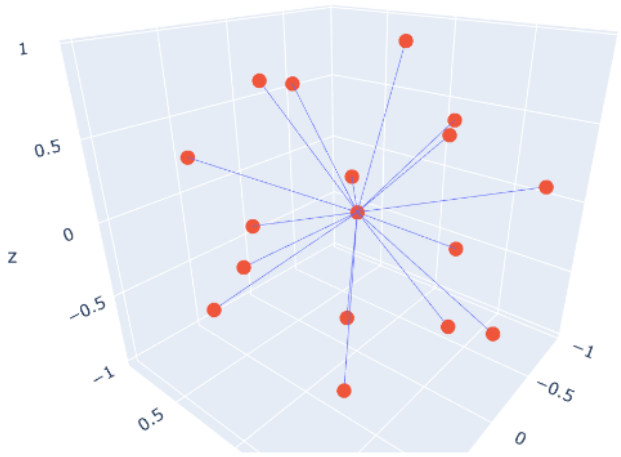

Figure 27 3D Subgraph Representation 2

5.5 Discussion Trees Analysis

After familiarizing ourselves with the underlying dataset and data structure, we focus on the properties that can be extracted from these discussion trees, starting with the obvious ones: depth and breadth.

5.5.1 Depth

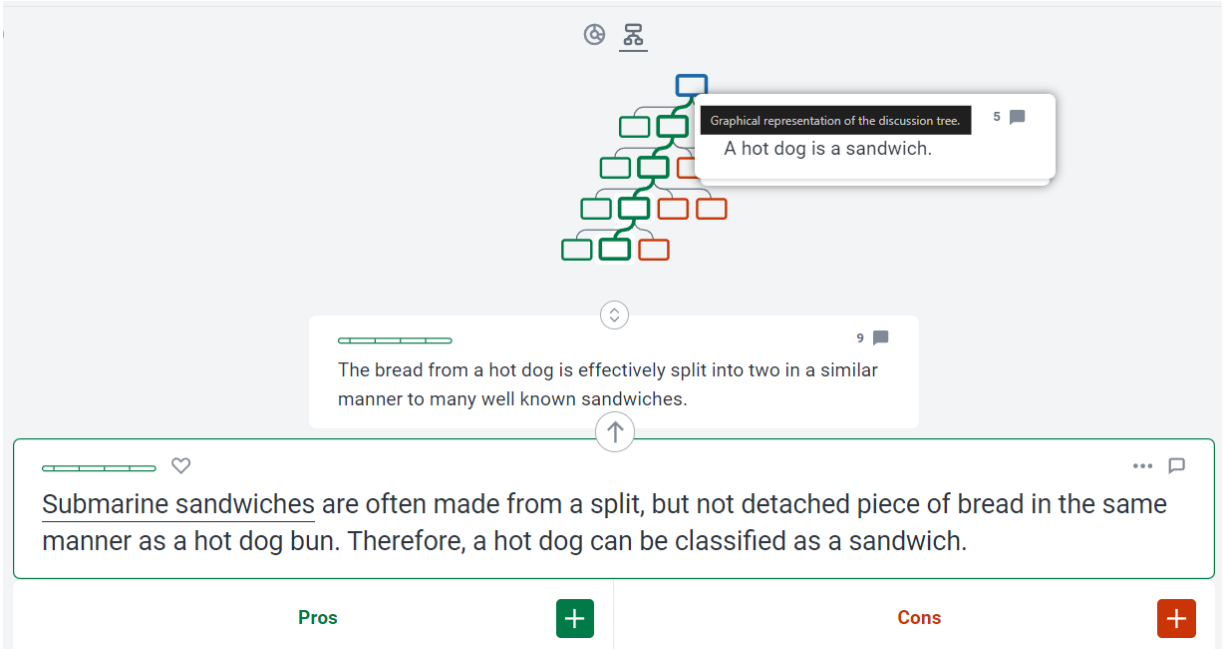

Figure 28 - Discussion Tree from Kialo: Is a hotdog a sandwich? (depth = 4)

- Depth gauges the "length" of the discussion threads. In a graph, this corresponds to the longest path from the central node (root claim) to any leaf node (endpoint) in a given theme or discussion. Nevertheless, what could that mean in a more linguistic sense? Observing several Kialo discussions makes it hard to say what depth stands for. However, some tendencies can be observed:
    - **High Depth**: Discussions with a high depth involve multiple layers of arguments and counterarguments. They likely represent complex topics where participants engage in detailed debate. **Possible reasons for this may be**:

  - Controversial subjects that require thorough discussion.
  - Topics which are strongly interconnected to other topics
  - Participants with solide expertise.
  - Complex and contentious topics, such as politics or moral philosophy, often have a high depth because they invite layered argumentation and counter-argumentation.

- **Low Depth**: Shallow discussions indicate straightforward exchanges with fewer layers of counterarguments. **Possible reasons for this may be**:
  - Topics may be less controversial.
  - Participants may reach consensus quickly.
  - Discussions on topics where there is widespread consensus or which are fact-based (e.g., specific scientific facts) might not go very deep, as there's little room for debate.

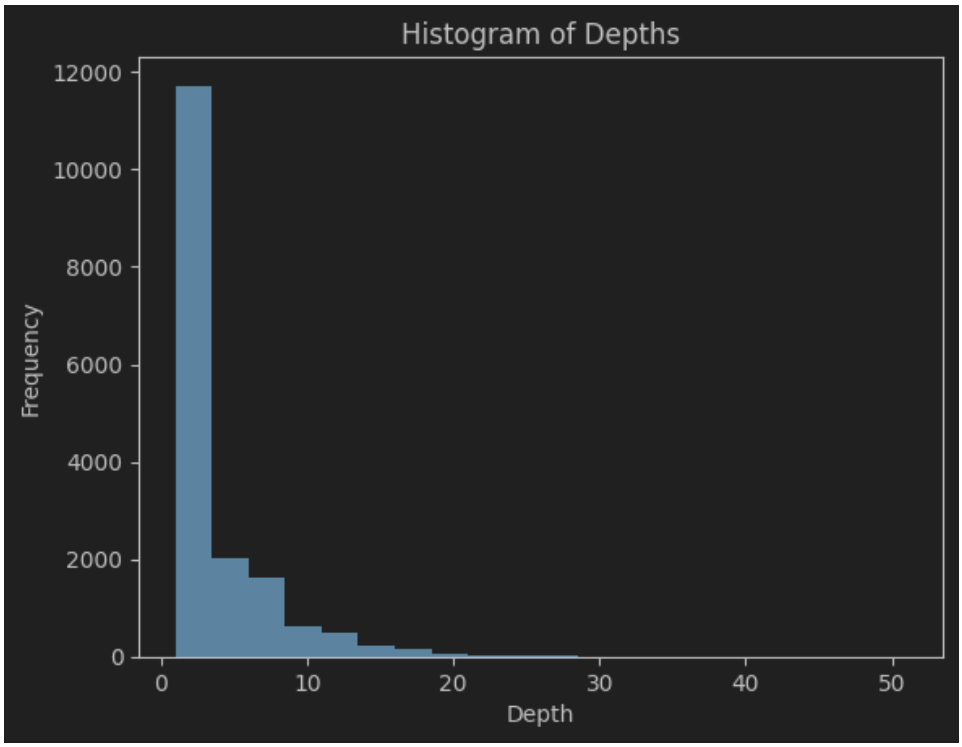

Figure 29 - Histogram of Discussion Depths

The histogram indicates the distribution of the depth of discussions. Most discussions have a low to moderate depth, indicating that extensive counterargument chains are less common.

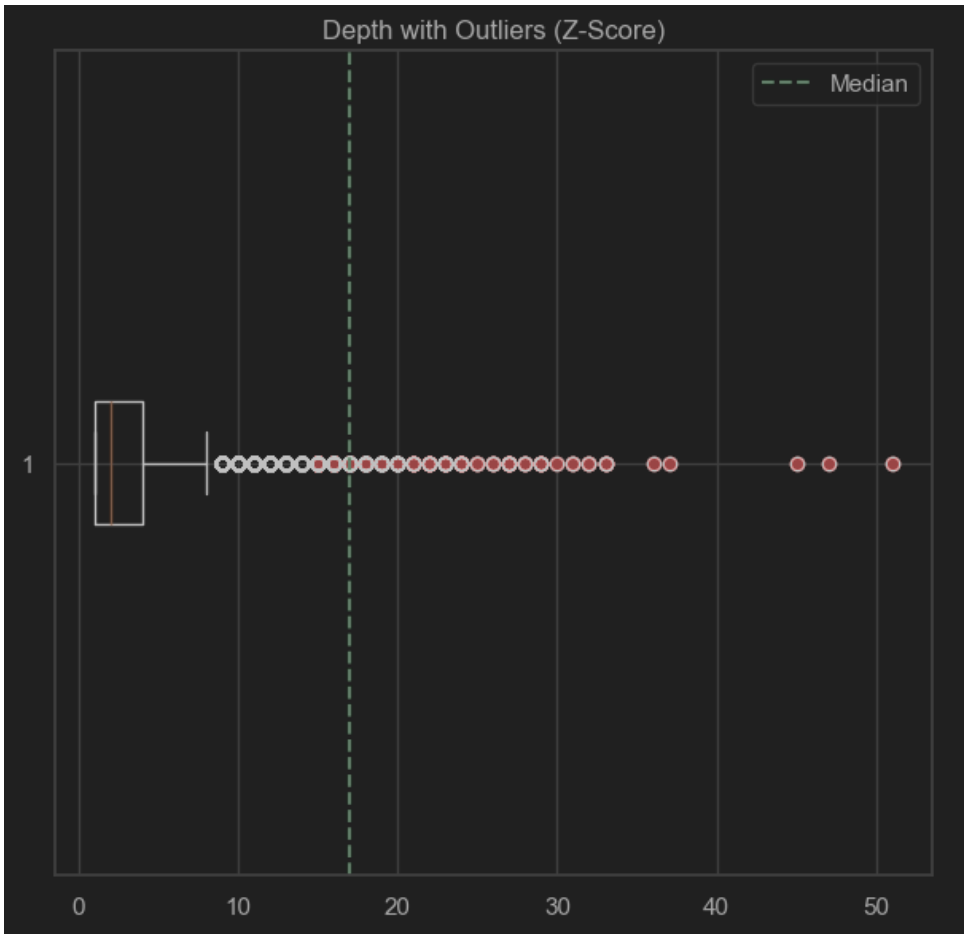

Figure 30 - Depth Outliers (Z-Score)

The median depth, indicated by the dashed green line, shows that while most discussions are shallow, there are significant exceptions where the depth of discussion is much greater. These outliers might represent complex debates requiring LLMs to process and generate responses that consider multiple layers of reasoning.

Few discussions have a depth significantly above the z-score median, suggesting topics or discussions where participants are highly engaged and provide extensive reasoning.

5.5.2 *Breadth*

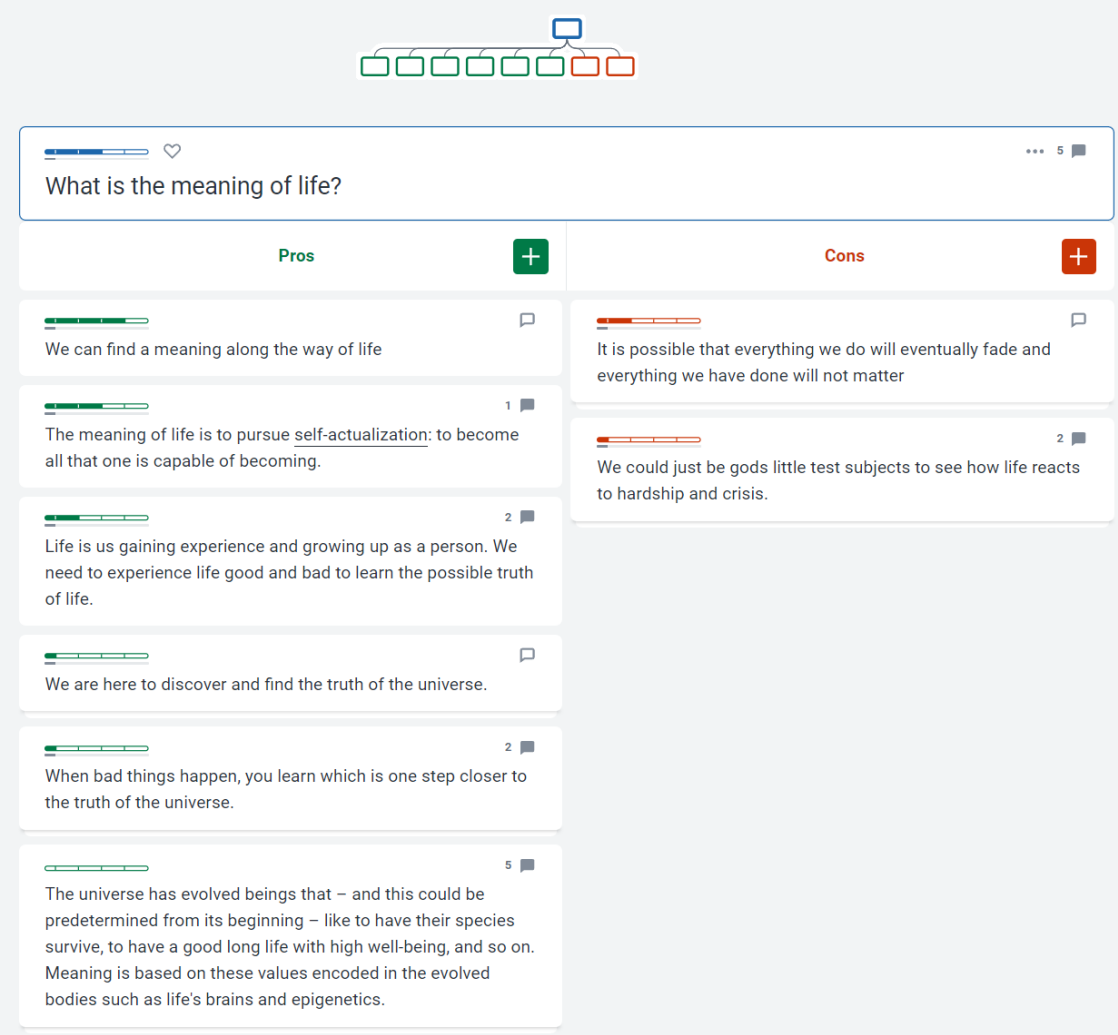

Figure 31 - Discussion Tree from Kialo: What is the meaning of life? (breadth=8)

- **High Breadth**: A high breadth suggests a variety of viewpoints at the same level of discussion, indicating a wide-ranging exploration of the topic. **Possible reasons for this may be** :
  - The topic has many subtopics.
  - The discussion is open-ended, allowing for broad participation.
  - Participants with different viewpoints (e.g., from different cultures, backgrounds)

  - General interest topics or those with broad societal impact, like sports events, major political decisions, or ethical dilemmas, might show wide breadth due to the diversity of opinions and the larger, more varied audience they attract.

- **Low Breadth**: Narrow breadth points to less diversity in viewpoints, potentially showing agreement or a focused topic with limited angles explored. **Possible reasons for this may be** :
  - A topic that is already well known or is similar to one already well known.
  - Majority of participants agrees.
  - Participants with similar viewpoints.
  - Topics that are highly specialized or technical, such as specific discussions within a field of science or a niche interest, might exhibit a narrow breadth as they attract fewer viewpoints or a more homogenous group of discussants.
- **Low Breadth and Low Depth** :
  - Unpopular topics
  - Nonrelevant topics
  - Discussions in early stages

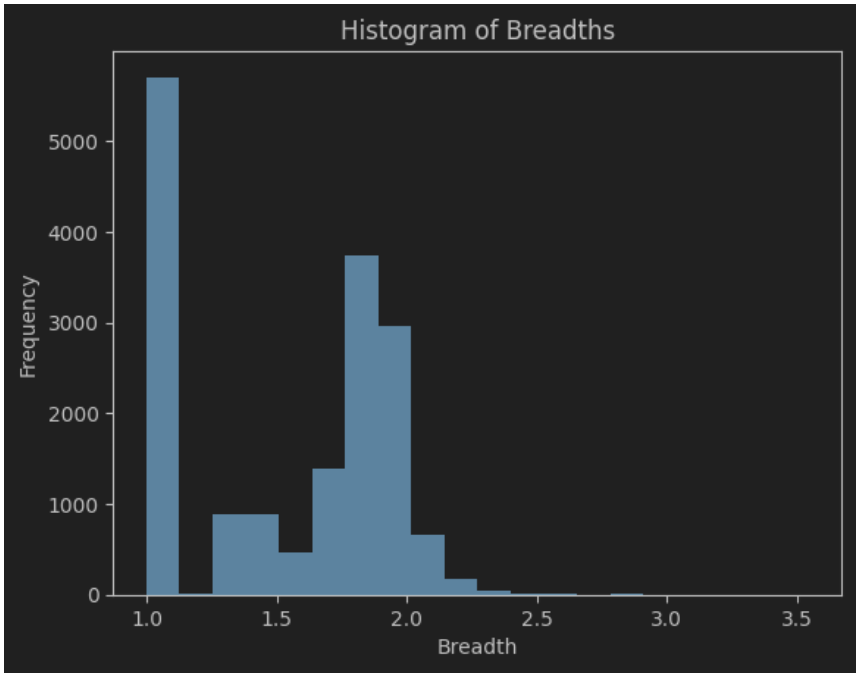

Figure 32 -HIstogram of Average Discussion Breadth

The data skews towards the lower end of the scale, indicating that many discussions have fewer unique viewpoints or arguments at each level.

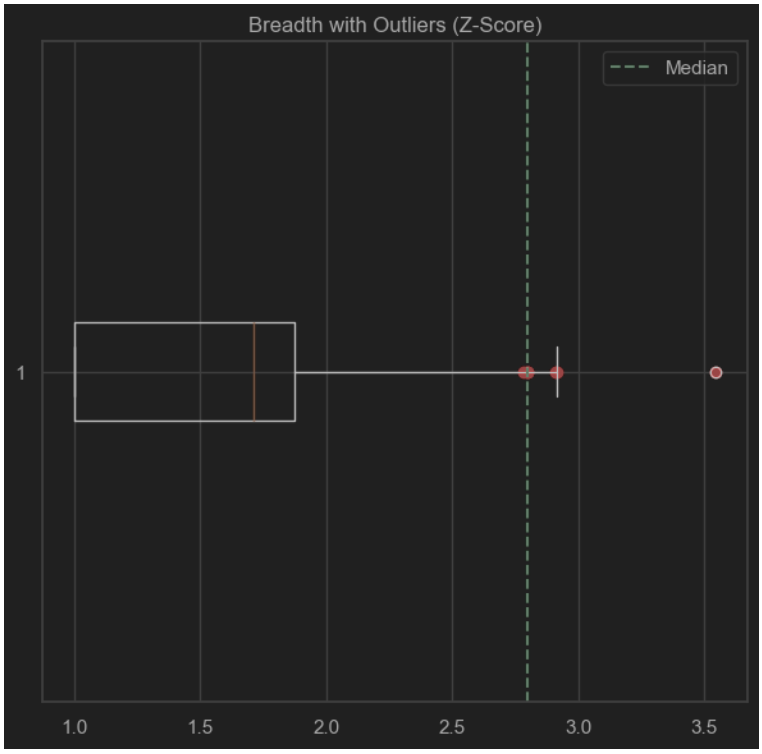

Figure 33 - Breadth with Outliers (Z-Score)

The breadth boxplot has a similar concentration around the lower quartile, with a small number of discussions being outliers with a higher breadth. These outliers signify discussions with a more excellent range of perspectives at specific levels of argumentation.

5.5.3 Breadth and Depth

While there may be tendencies for certain categories to have specific depths or breadths, it is not a strict rule. The depth and breadth are also influenced by the specific question at hand, the participants involved, the moderation of the discussion, and the cultural or temporal context. It's possible for a scientific topic to have a deep and broad discussion if it touches on controversial or emerging theories. Conversely, a political topic might have a narrow and shallow discussion if the focus is on a universally accepted principle or a well-established fact.

Correlation analysis of breadth and depth numbers per discussion show a value of 0.66 indicating a moderate positive relationship, s suggesting that discussions also tend to have a broader range of viewpoints and vice versa as discussions deepen.

5.5.4 Centrality and Communities

For exploring tendencies within the node data frame, centrality measures are used, which are apt for discovering graph structures. They help estimate the discussion data available and serve as an approximate frame/border for generating the tasks. They must be understood as an estimation and only a definite fact once they are calculated over all claims. The following histograms observe the interconnectedness, closeness, and distributions of claims.

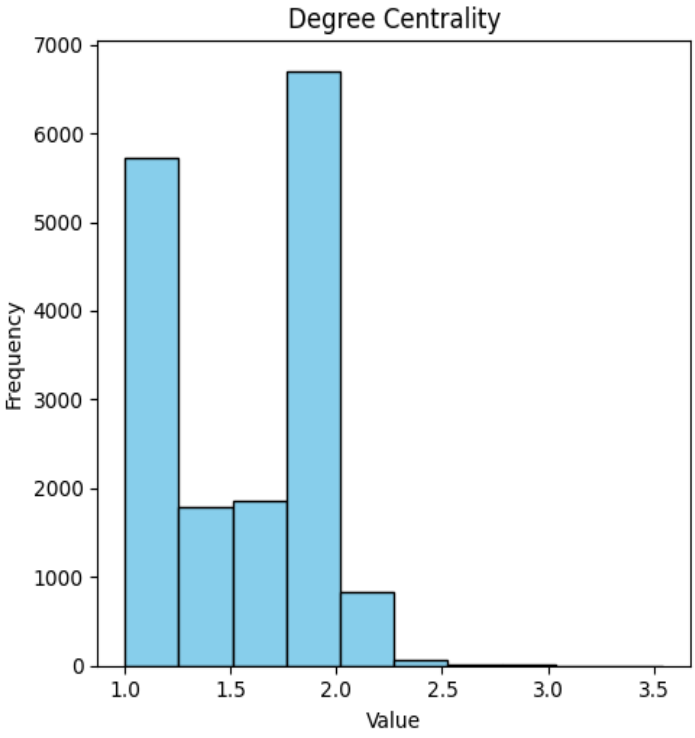

Figure 34 - Histograms: Degree Centrality

- The histogram for degree centrality shows two primary peaks around the values 1.0 and 2.0, indicating a significant number of nodes with these degree centrality values. The median degree centrality of 1.714286 suggests that the average node is connected to slightly more than one other node.
- This pattern suggests that while there are nodes with connections (indicative of active participation), a majority of nodes have limited interactions within the network.

- **Closeness Centrality**

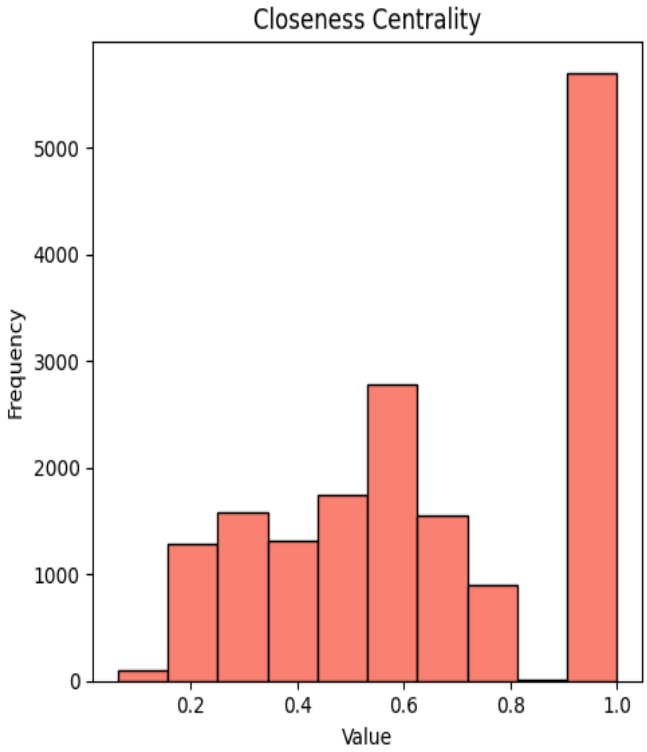

Figure 35 - Histogram for Closeness Centrality

- The closeness centrality histogram displays a distribution skewed towards higher values, with a significant peak close to 1.0. The median closeness centrality of 0.596154 indicates that over half of the nodes have a closeness centrality only little less than 0.6 (threshold for being considered "Central" in their respective discussions).
- A node is considered "Peripheral" if its closeness centrality is less than 0.4. In the boxplot and the median, we observe that more values are greater than this value , meaning that most nodes/claims lie within the discussion and not on the outer side.
- The high number of claims around 1.0 could represent the root claims which often have the most directly following claims. So, it is fast at another claim. Also, we count around 7'840 discussions which is not that far from the barplot at 1.0.

Betweenness Centrality

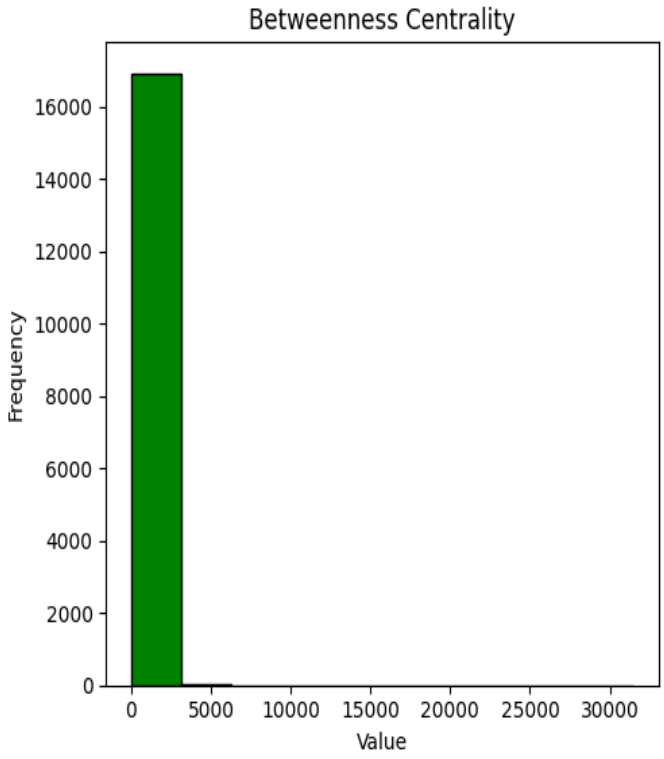

Figure 36 - Histogram for Betweenness Centrality

- The betweenness centrality histogram reveals a high number of nodes with a low betweenness centrality value, close to 0. The median value of 2.142857 is also low, indicating that most nodes in the network do not act as bridges between other nodes.
- This observation aligns with a network where few nodes play a significant role in connecting different parts of the discussion, which are the root nodes of each discussion.
- A node is considered a "Key bridge" if its betweenness centrality is greater than 1/4 N^2, and "Less central" if it's less than 1/8 N^2.
- Given N=17830 the threshold for being a "Key bridge" would be $1/4 \times 17830^2 =$ 794´772´25 and for "Less central" it would be half that, approximately 397´386´12.5.

- Many nodes in the histogram have a betweenness centrality value much less than 39′731′926.5. This indicates that few nodes serve as key bridges in the network, aligning with the findings that there are not many deep connections between claims.

Communities

The power-law distribution of community sizes indicates that the Kialo dataset is characterized by numerous small, specialized discussion threads with only a few larger, more interconnected ones. This suggests a wide variety of topics with many discussions being insular and possibly centered around specific or niche subjects.

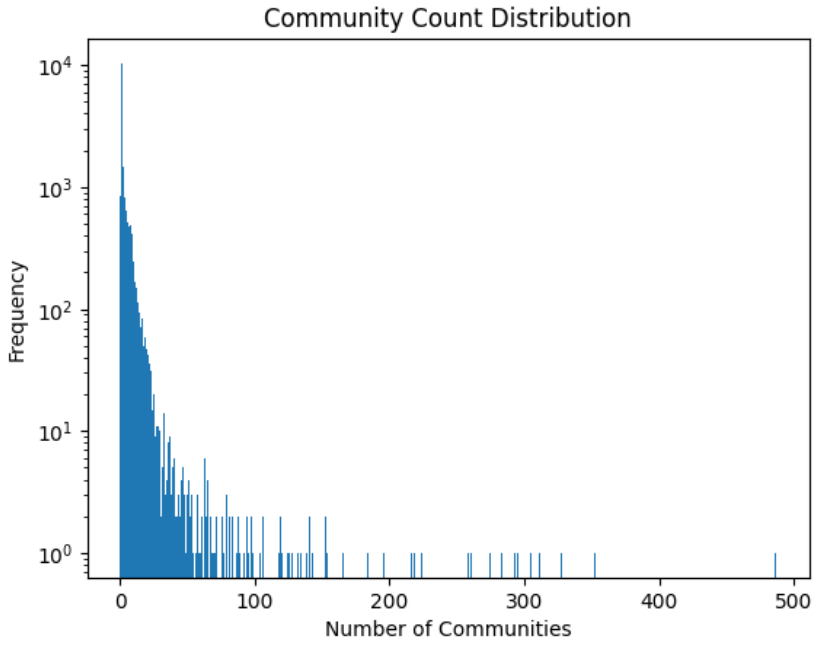

Figure 37 - Community Distribution

Most groups of claims have low external connectivity due to the nature of isolated main themes in Kialo and discussions not always going into deep debates.

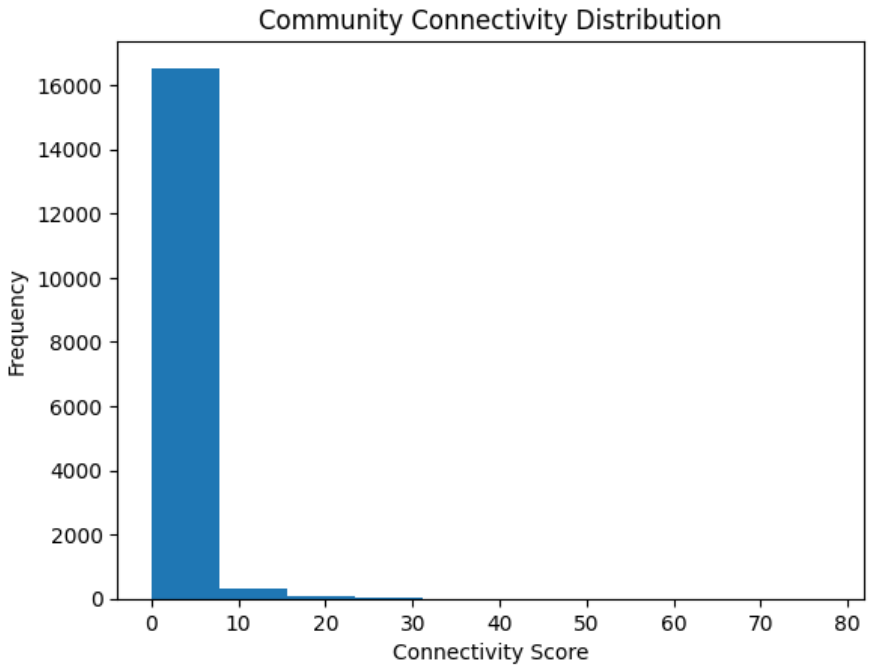

Figure 38 - Community Connectivity Distribution

In summary, the data is expected to show a landscape of diverse and fragmented discussions, with most being self-contained and a few more expansive and connected.

5.5.5 Disconnected Subgraph - Conversation Analysis

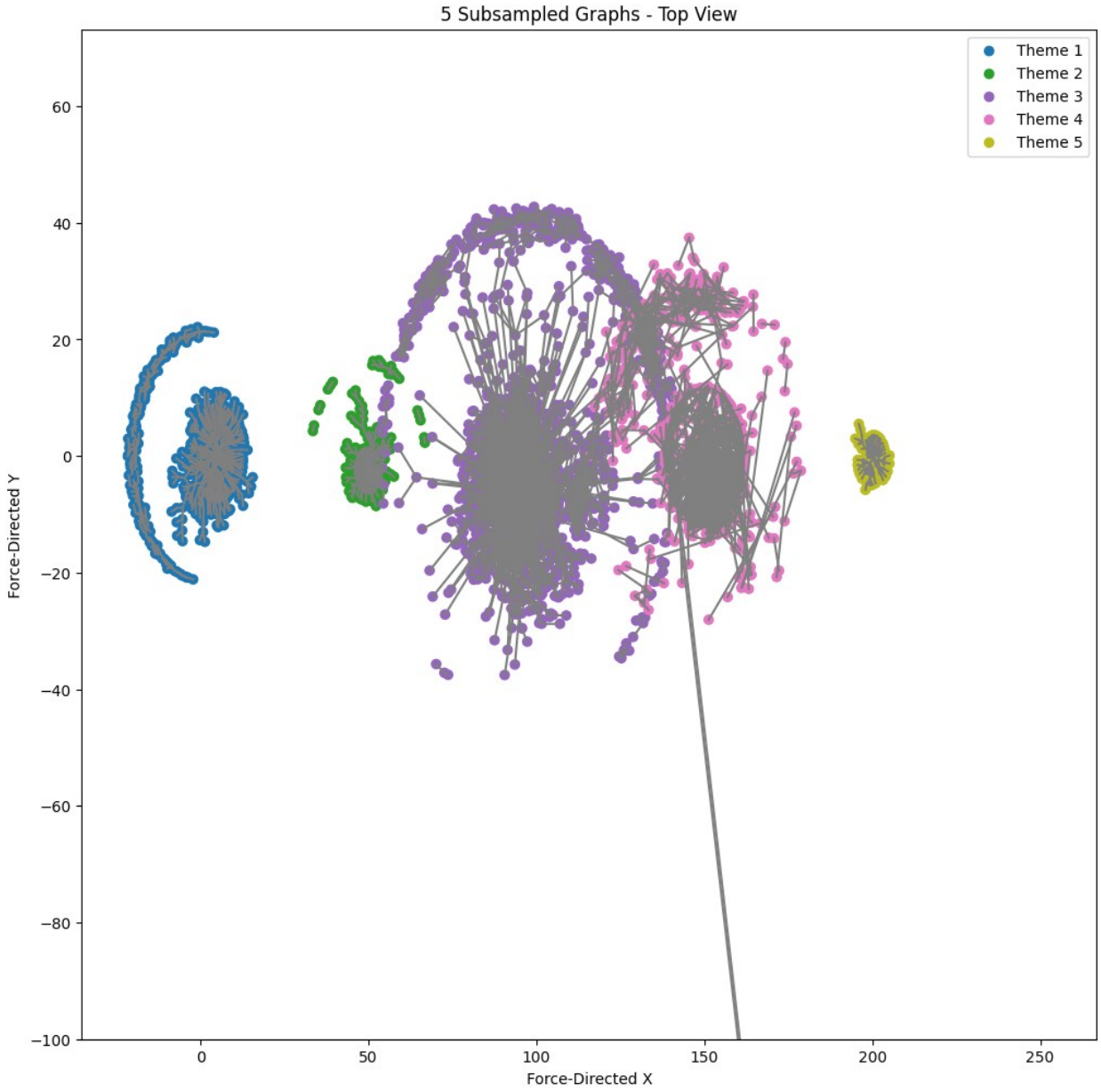

Figure 39 - Deviating Discussions

In the visualization of discussion graph subsamples (again the axis are just labelled because the Kamada-Kaway algorithm is used for more aesthetic visualization), a distinct a line is visibly diverging from the main cluster of subgraphs. Further analysis reveals that this represented a

disconnected subgraph, typically indicative of claims that stray from the central theme of the discussion.

The primary subgraph consisted of conversations revolving around the utility of an anonymous, untrackable digital currency, such as Bitcoin, and its relevance to a modern society:

- *Anonymous currency discussions*

  An anonymous untrackable digital currency \(ADC\), like Bitcoin, is beneficial for civilized societies.

- *Cash is a necessary part of any functional society's economic model and cryptocurrencies are an easier and online-ready form of cash. They are the natural progression of what money is becoming in the digital age.*

- *Most ordinary citizens would not use such a currency anyway, rendering all potential benefits moot.*

  Most people pay taxes and banks report all their in/outflows

The dialogue included discussions on the evolution of money in the digital age, the necessity of cash in economic models, and the likelihood of widespread adoption by the general public, considering existing financial regulations.

In contrast, the disconnected subgraph took a tangential path. The dialogue within this subgraph shifted focus from the societal implications of cryptocurrencies to specific individual scenarios involving financial privacy and cultural perceptions. This outlier from the main conversation thread showcased a shift to personal experiences and societal stereotypes, diverging from the primary discussion on anonymous digital currencies and their systemic impact.

- *Then be a man and make sure that she does not see that credit card bill. You even have the present as a perfect explanation.*

- *But that is suspicious, Tiffany will look at me in a very funny way.*
- *They are used to it from the nouveau riche from Eastern Europe and the Ex-USSR.*
- *But I do not want to appear like the Russians.*
- *Then make a deal with them that you will pay them later or wire the money from your bank account.*

This shows that a discussion can indeed branch out in subdiscussions and themes which are not in the context of the topic.

5.6 Probing Tasks

The discussion trees were explored in the previous Chapter, and key features were identified using the parsed discussion trees. The actual implementation of the probing tasks starts according to the derived definitions in Section 5.2. Let´s review them in an abbreviated version focusing the tree properties that are being processed in the following sections:

1. **Stance Alignment**: Probes the model's capability to detect the stance of two claims.
2. **Sequential Coherence**: Probes the model's capability to predict whether two claims were directly responding to each other.
3. **Interactive Dynamics**: Probes the model's capability to predict interaction dynamics, such as the expected number of responses to a claim.
4. **Claim Depth Hierarchy** Probes the model's capability to predict the hierarchical structure (depth level) of two claims.
5. **Discussion Contour Recognition**: Probes the model's capability to predict the number of outgoing claims and depth levels between two claims.

5.6.1 Data Preparation

To generate the probing data, a data frame named **node_df** is created that allows fast access to the linguistic key features for each node such as depth, breadth, previous node, and root node. This data frame has been structured to allow quick access to features necessary for the tasks conceptualized in Section 3.3. The **node_df** is denormalized, enhancing efficiency for task-related operations. It includes the following features:

- **Node_ID**: This unique identifier is assigned to each claim within a discussion, typically generated sequentially as claims are processed.
- **Depth**: This attribute measures how many layers deep a claim is within the discussion tree. It starts at 0 for the base claim and increments by 1 for each subsequent level of the discussion. The depth is calculated using a process akin to a depth-first search, where each node's depth is determined by how many edges it is removed from the root node.
- **Breadth**: This attribute assigns an integer to enumerate each claim at a specific depth level. It represents the order in which nodes are discovered within their level, as determined by a breadth-first search through the discussion tree.
- **Parent Node**: It points to the ID of the claim that directly precedes the current claim in the discussion, creating a link in the tree hierarchy and indicating the flow of the discussion.
- **Base_Claim_ID** and **Base_Claim_Text**: These fields hold the identifier and text content of the root claim of the discussion, anchoring all subsequent claims to their origin topic.
- **Claim_Text**: This is the text of the current claim being examined or processed.
- **Outgoing_Node_IDs**: A list of Node_IDs that the current claim directly responds to or follows from, providing a way to traverse the discussion tree.

- **Relations_To_Outgoing**: Alongside each outgoing node ID, this list indicates the type of stance (1.0, -1.0.) the current claim has with its directly outgoing claims.

5.6.2 General Process

For the creation of the probing tasks the same high-level process is used leading to a more efficient workflow and ensuring compatibility with the probing framework:

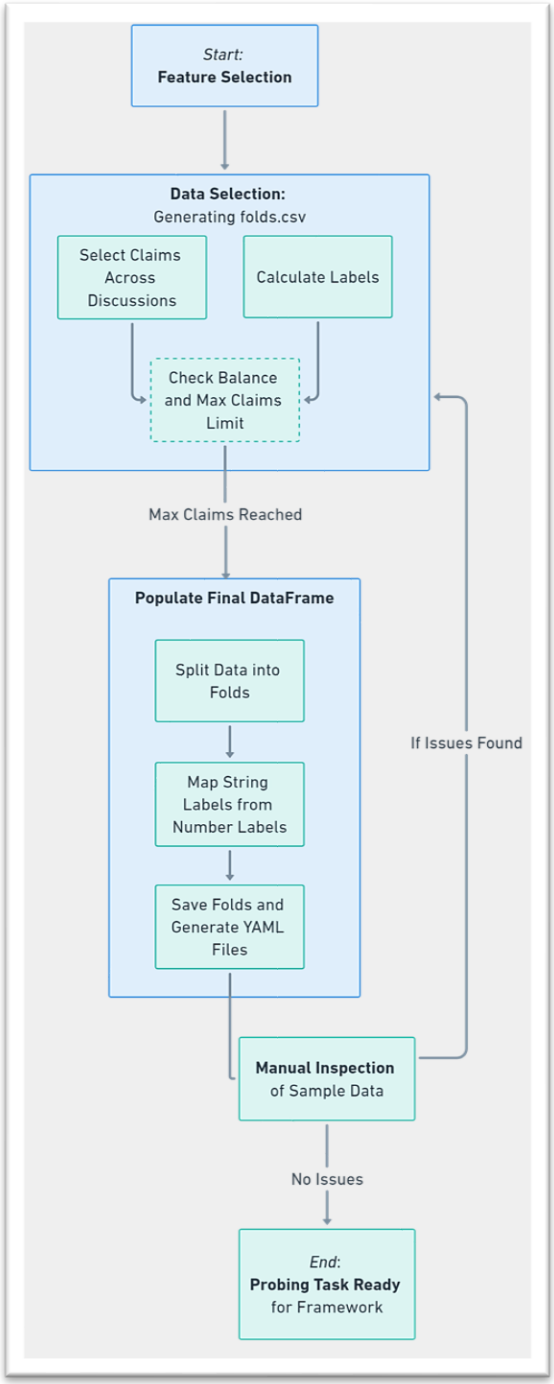

Figure 40 - Probing Task Design: High Level

1. **Feature Selection**: The process begins with the selection of a specific discussion tree features that has been verified in the data analysis. It is chosen for its ability to reflect a specific linguistic property as defined in Section 5.2.
2. **Data Selection**: The next step involves generating a data frame, which later is probed as **folds.csv** in Juypterlab. The process encompasses two main activities: selecting claims from various discussions (input claims) and calculating labels for these claims. The aim is to ensure a balanced representation of labels across discussions until a maximum limit is reached (mostly 100´000 probing examples, sometimes 200´000).
3. **Populate Final DataFrame**: Once the data selection is complete and the maximum number of claims is reached, the focus shifts to populating the final data frame. This step involves 1.) splitting the data into 4 folds preventing selection bias as described in Section 4.7.2, 2.) mapping string labels from number labels, and 3.) saving of the folds and generation of YAML files that are ready to be run on Juypterlab. Each input column is rechecked for compatibility with the probing framework using the Python function **ast.literal_eval,** which throws an error if the claim input strings are formatted incorrectly (for example, in the case of an unclosed bracket).
4. **Manual Inspection**: After the final data frame is prepared, a manual inspection of a sample of data (approximately 30 examples per fold) is conducted. This step is essential to ensure the quality and accuracy of the data. If any issues are identified during the inspection, the process reverts to the data selection step for necessary adjustments. This method does not guarantee that every example created is meaningful. However, manually inspecting 100´000 rows of data is practically impossible in the given time frame. To know if the probe was meaningful enough for the LLMs to solve, it must be tried.

5.6.3 Data Cleaning

In the data cleaning process for this project, several steps are implemented to enhance the quality and relevance of the data before generating probing tasks. The primary objective is to remove irrelevant or incorrect information that could confuse the probed LLMs and to standardize the data format for analysis. These steps are presented in no specific order, as only some data issues were apparent from the start. For instance, spam messages became noticeable empirically only during the development of probing task 1. The key steps include:

- **Processing Website Links**: Extracts meaningful parts from URLs, omitting unnecessary elements like 'https://' for simplicity.
- **Text Preprocessing with Stopwords Removal**: Removes stopwords and punctuation and converts text to lowercase, enhancing clarity and uniformity.
- **Non-English Sentence Identification and Removal**: Utilizes the **langdetect** library to detect and exclude non-English sentences, ensuring language consistency.
- **Spam Detection and Elimination**: Employs regex patterns and additional criteria, like capitalized word checks, to identify and remove spam content, preserving data integrity.
- **Duplicate Discussion Removal**: Checks for and removes duplicate discussions to prevent data redundancy and maintain dataset diversity.
- **URL and Special Character Handling**: Corrects issues with unclosed brackets and special characters resulting from URL cleaning, ensuring text formatting accuracy.
- **Conversion of IDs to Numeric Format**: Transforms text-based IDs to a numeric format for standardized data processing.

5.6.4 Probe 1: Stance Alignment

For Probe 1, the core feature utilized for generating the task is the relational structure inherent in the edges of the discussion tree. The LLM is being probed on two input claims and has to find out if they affirm or oppose each other.

Data Selection Process in generate_stance_data

1. **Prepare Data:** Group the data by root claim (Kialo topic).
2. **Iterate Over Discussions:** For each discussion:
    - Skip discussions with fewer than three nodes.
    - Select the root claim (**starting_claim_row**).
    - Determine the maximum depth of the discussion.
3. **Generate examples:** For a set number of trials (up to **max_trials**):
    a. Initialize two paths for random walks, starting from the root claim.
    b. For each path:
        i. Randomly select a target depth within the discussion.
        ii. Call the **random_walk** function to navigate the discussion tree and return the final claim and stance.
    c. Calculate the final stance as the label by multiplying stances from both paths. .
    For example:-1-1 = 1*
    d. Form a pair of final unique claims as inputs.
    e. End trials if the maximum number of examples (**max_examples**) is reached or all trials are completed.

Find Target Claim in random_walk

Walk the Tree: For each depth level up to the target depth:

a. Skip if there are no outgoing relations or if there are no further claims.
b. Collect the current claim ID and stance.
c. Select a claim and stance randomly from the available outgoing claims.
d. Update the current stance based on the chosen stance.
e. Proceed to the next claim in the path.
f. Stop if no further claims are available or an empty path is encountered.

Integration of Both Functions

- **generate_stance_data** creates unique stance pairs by performing two independent random walks (using **random_walk**) for each discussion in the dataset.
- **random_walk** function is used to simulate a path through the discussion, capturing the evolution of claims and their stances, which are then used to form stance pairs in **generate_stance_data**

Moving from Version 1 to Version 2: ***Stance Alignment light***, the simplifications are:

- **Labels**: No longer derived from multiplicative stances. Now, only sequential claims are selected, and labels of their immediate relationship are directly assigned to them removing long range dependencies.
- **Inputs**: Random walks are removed, therefore reducing variability.

inputs	topics	org_label	label
('Love is rational, as it is a consequence of your unconscious.', 'Love is a result of having high levels of some hormones and neurotransmitters in our body.')	Love is rational, not irrational	Confirming	1
('Love is a result of having high levels of some hormones and neurotransmitters in our body.', "Love is irrational because you can't just choose who you want to love.")	Love is rational, not irrational	Opposing	-1

5.6.5 Probing 2: Sequential Coherence

This task aims to assess how well a language model can recognize the sequential coherence between claims in a discussion. This task centers on distinguishing claims that are immediately connected (sequential) from those that are separated in the discussion flow.

1. The data selection process for this task involves the following steps, with a focus on the inputs and labels:

1. **Group Data by Discussion** to process each discussion individually.
2. **Identify Immediate and Separate Claims**:
    - **Immediate Claims**: Each claim in a discussion identifies its parent node. If a claim directly follows its parent in the discussion, it is considered an "immediate" claim. The pair of the parent claim and the current claim forms an 'Immediate' pair.
    - **Separate Claims**: Claims that do not directly follow their parent node are considered "separate." A separate claim is randomly selected and paired with the current claim to form a 'Separate' pair.

3. **Label Assignment**:
    - Each 'Immediate' pair is assigned a label of '1', indicating a sequential or direct connection.
    - Each 'Separate' pair is labeled '0', indicating a lack of immediate sequential connection.
4. **Balance and Shuffle Pairs**: The task strives to create a balanced dataset of 'Immediate' and 'Separate' claim pairs. The pairs are then shuffled to randomize their order.

inputs	topics	org_label	label
'In cities, dogs can pose a health risk.' , 'Many dogs don't have the space they need in cities. '	Who will defeat the Night's King?	Immediate	1
('The UN and NATO have not historically done enough.', 'They have barely responded to the Russian invasion of Crimea.')	Would we like to live in the world of Harry Potter?	Immediate	1
('Security cameras do not stop violence; only record it.', 'Children our our most valuable resources. Schools should absolutely allocate more resources toward stronger security.')	Jesus probably did not exist as an historical person	Separate	0

5.6.6 Probing 3: Reactiveness

This is the only probe that only has one claim as input. The model needs to predict the reactiveness level of that claim in some discussion, granularized as "Strong," "Moderate," or "Weak," based on the number of subsequent claims related to it.

Implementation Steps:

1. **Calculate Reactiveness Score**: Each claim's reactiveness score is calculated by counting the number of outgoing claims that directly follow it.
2. **Normalization**: The reactiveness scores are normalized within each discussion by dividing by the max score per discussion. This normalization accounts for varying sizes of discussions, ensuring a fair comparison of reactiveness scores across different contexts.
3. **Label Generation**: Reactiveness scores are dynamically labeled using percentile distribution. Claims with reactiveness scores in the top third percentile are labeled as "Strong," those in the middle third as "Moderate," and the bottom third as "Weak." This approach ensures a balanced representation of different levels of reactiveness in the dataset. This could be made more challenging when asking the model to predict the exact number / using more labels.

inputs	topics	org_label	label
('The top 1% own 40 percent of US wealth, while the bottom 80% own only 7 percent',)	The cost of Education, Healthcare, Home Ownership and Military Spending are means of fleecing the wealth of the masses for transfer to the top 1%.human creativity	Strong	2
'Humans cannot digest (see website: chemistry)'	What is the best diet or dietary approach for health and well being?	Weak	0

5.6.7 Probing 4: Claim Depth Hierarchy

For Task 4, we aim to assess a language model's ability to distinguish between claims at the same or different depths within a discussion. This task aims to understand how well the model can identify the relational hierarchy in discussions.

Data Preparation and Generation

1. **Group by Root Claims:**
2. **Random Claim Pair Selection (draw_random_claims Function):**
    - Randomly picks pairs of claims within each discussion from the depth levels defined in the variables **same_level_criteria = {'Depth': 1}** or **diff_level_criteria = {'Depth': 0}**. By increasing the depth level the claims are sampled from deeper discussion levels and are estimated to represent more complex claims.
3. **Generate Examples (generate_examples Function):**
    - Iteratively creates claim pairs using draw_random_claims.
    - Balances same and different level claim pairs.
    - Forms inputs from claim pairs, with labels derived from their depth relations (1: “same level”, 0:”different level”).

The light version generates positive examples by ensuring the claim pairs share a common parent node. This criterion simplifies the process by focusing on a more direct and evident relational link between claims. Generating negative examples (claims at different levels) remains the same.

Generate Light Examples (generate_examples_light Function):

- Iterates over each discussion, identified by unique base claim IDs.

- Generates positive examples from claim pairs at the same level, ensuring they share a common parent node. This simplifies the task by focusing on a more direct and evident relational link between claims.
- Produces an equal number of negative examples from claim pairs at different levels, reflecting diverse discussion structures.
- Balances the dataset with an equal number of positive (same level) and negative (different level) examples.

Main Function (main_function_4_light):

- Calls generate_examples_light to create the dataset.
- Prepares and structures the final dataset for model training, ensuring each example is appropriately labeled and categorized.
- After generating examples, the dataset size is significantly reduced. This reduction involves filtering out empty examples and limiting the number of examples to 10 positive and 10 negative examples per discussion topic.

inputs	topics	org_label	label
'It is unique', 'It is not unique'	"Millennial burnout" is not unique to the millennial generation	Same Level	1
'3-D printers will save the world.', 'Food can be printed for those who live malnourished.'	3-D printers will save the world.	Different Level	0

5.6.8 Probing 5 - Discourse Contour Recognition

For Probing Task 5, the objective is to evaluate a language model's understanding of discussion complexity by focusing on the average number of possible directions (outgoing nodes) per depth level in a discussion. This is quantified by a ratio representing the average branching complexity at each depth.

Data Preparation and Analysis Process:

1. **Unique Depth Identification**: The function **generate_examples** starts by grouping the dataset by the unique base claim IDs to separate discussions.
   It then identifies unique depth levels within each discussion, excluding the root (depth=0) to focus on the sub-discussions.
2. **Random Depth Selection and Ratio Calculation**:
   - The algorithm randomly selects a depth level within the discussion, not including the maximum depth to ensure there are subsequent levels to analyse.
   - It calculates the ratio of possible outgoing nodes (directions a discussion could branch) to the depth levels traversed. This ratio is computed as **num_of_claims_below / depth_levels_traveled**, where **num_of_claims_below** is the cumulative count of outgoing nodes from the selected claim to root claim upwards and **depth_levels_traveled** is the count of depth levels traversed from the selected claim to the root node. The root node is reached if there is no parent node (previous node) left for the selected claim so the while-loop breaks.
3. **Example Generation**:
   - For each pair of claims identified at the chosen depth level, their texts are paired along with the calculated ratio.

- This ratio measures how many branching options are available on average at a given depth ~~in the discussion~~. A higher ratio indicates a more complex, branching discussion structure, while a lower ratio suggests a more linear discussion flow.

4. **Label Assignment Based on Ratio**:
    - Labels like "Sparse," "Moderate," "Dense," and "Very Dense" are dynamically assigned based on the calculated ratio. These labels are determined by percentile thresholds (25th, 50th, 75th) of the ratio distribution in the dataset.

inputs	topics	org_label	label
('Dogs do not have souls therefore they cannot have emotions.', 'According to Buddhism, animals have souls.')	Dogs do not have complex emotions such as shame, guilt, and pride.	Dense	2
'Relying too heavily on technology makes a society vulnerable.', 'Technology has contributed to a better medical care.'	The NHS should be privatised	Very Dense	3
('God allows evil to exist so that humans can have free will and develop into moral people. In order to be morally good in a meaningful way, a person must have the possibility of choosing evil.', 'If God gave us free will, evil could be a rebellious act against God, so in this sense God and evil could coexist.')	Because of the existence of evil, there cannot be a monotheistic God \(as traditionally conceived\).	Sparse	0

5.6.9 Distribution Summary of Probing Task Data

New Probe Name	Number of unique topics	Number of discussions	Average number of discussions per topic	Median number of discussions per topic	Standard deviation of discussions per topic	Number of unique labels	Label distribution	Number of outlier discussions (z-Score)	Range of topic counts per Discussion
"1 Stance Alignment"	3694	175497	47.51	42.5	38.55	2	"{-1: 47 , 1: 52}"	68	2 - 279
"1_5 Stance Alignment light"	6605	118112	17.88	11.0	14.47	2	"{0: 48 , 1: 51}"	13	1 - 120
"2 Sequential Coherence"	546	200000	366.30	72.0	976.59	2	"{0: 50 , 1: 49}"	12	3 - 9086
"3 Interactive Dynamics"	6809	233794	34.34	11.0	87.11	3	"{0: 62, 1: 5 , 2: 31}"	86	1 - 4252
"4 Claim Depth Hierarchy"	4224	112601	26.66	27.0	11.36	2	"{0: 49 , 1: 50}"	27	1 - 181
"4_5 Claim Depth Hierarchy light"	2399	100000	36.86	21.0	45.34	4	"{0: 22, 1: 22, 2: 28, 3: 26}"	0	1 - 20
"5 Discourse Contour Recognition"	2463	100000	40.60	23.0	51.48	4	"N/A"	1	1 - 11572

5.6.10 Juypterlab

For running the probing data, the finalized folds.csv, together with the probe-specific YAML configuration, are exported in a separate folder per task into the root folder ” probing_files” on Juypterlab where the probing framework was installed and prepared by the main advisor. This framework is run on an HSLU server, which allows the usage of GPU for timely probing results.

The YAML files have to be in a consistent format for the probing framework.

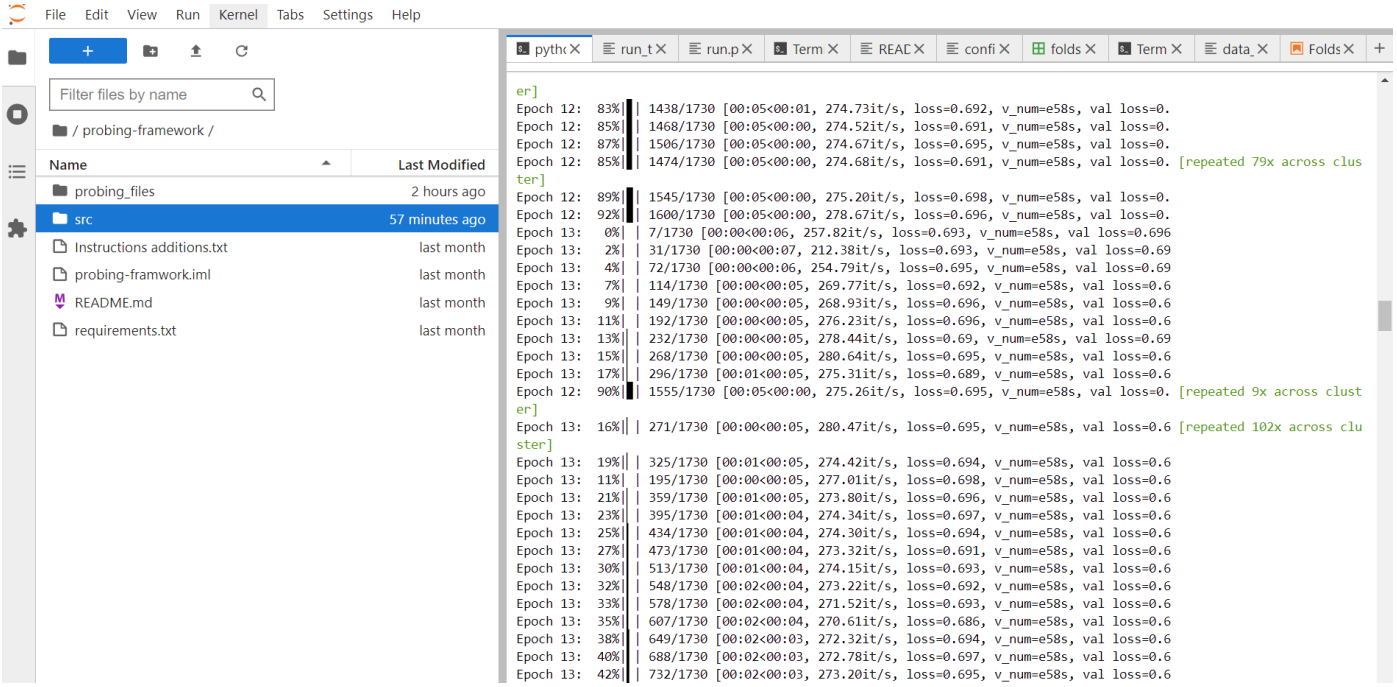

Figure 41 - Running Probing Task in Jupyterlab

1. **File Naming Convention**: The file names for the YAML configurations are systematically created to reflect the probing task's characteristics. They follow the pattern **config(-bi)-[controltask].yaml**, where:
    - **(-bi)** is appended if **num_inputs** equals '2', indicating that the probing task deals with pairs of inputs.
    - **[controltask]** part is replaced by either **none**, **perm**, or **rand** to indicate the type of control task applied (**NONE**, **PERMUTATION**, or **RANDOMIZATION**, respectively).
2. **Attributes within the YAML Files**:
    - **control_task_type**: Specifies the type of control task. Can be **NONE**, **RANDOMIZATION**, or **PERMUTATION**.

- **num_inputs**: Indicates the number of input statements used in the probing task. As per the code, the possible values are '1' or '2'. If **num_inputs** is '1', the **probe_task_type** is set to **SENTENCE**; otherwise, it is set to **SENTENCE_PAIR_BI**.
- **num_labels**: Defines the number of unique labels the model is expected to predict.
- **num_probe_folds**: Fixed at '4' across all tasks, representing the data split into four folds for the probe.
- **probe_name**: Constructed by combining the probe number with a descriptive name, it serves as an identifier for the specific probing task.
- **probe_task_type**: Determines the format of the probing task, which can be **SENTENCE** for single inputs or **SENTENCE_PAIR_BI** for pairs of inputs.
- **probes_samples_path**: Name of the directory path where the probing files are stored with the suffix "/probe_name".

```
control_task_type: NONE
num_inputs: 2
num_labels: 4
num_probe_folds: 4
probe_name: probe_5-claim_path_density_categorical
probe_task_type: SENTENCE_PAIR_BI
probes_samples_path: probing_files/probe_5-claim_path_density_categorical

```

Figure 42 - YAML configuration file

WANDB Logging

The LLMs runs were logged directly on „Weights And Biases" (WANDB). For every LLM, every probe 3(a separate project in WANDB) runs on 4 different folds and 5 different seeds. So,

in the end, there are 20 runs per model and probe group. Some probes were also run more than once. The runs were grouped by model name and control task in the image. It shows all the logged attributes of the aggregated (and averaged) runs (such as folds, seeds, batch size, GPU, runtime).

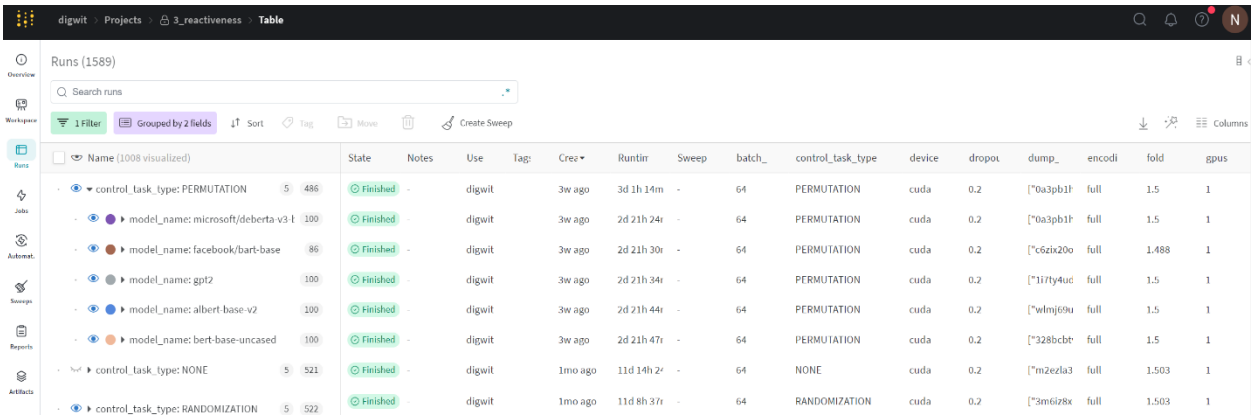

Figure 43 - WANDB Logging

Crashed Runs

Since the HSLU servers are not held in a completely isolated environment, maintenance works or fluctuations in the network can lead to crashed runs.

5.7 LLMs Performance Expectations

By considering again the findings and model cards from Hugginface from Chapter 2 this Section tries to build some expectations before evaluating LLMs.

5.7.1 Expectations Based on LLM Group Characteristics

BERT-related Models: BERT and its variants are expected to shine in scenarios requiring sophisticated context management due to their bidirectional nature, making them adept at interpreting the ebb and flow of discussions that align with human cognitive functions such as working memory and context processing. These models will likely have a propensity for tasks

that involve intricate contextual relationships, akin to the executive functions attributed to our prefrontal cortex.

GPT-related Models: GPT models, with their unidirectional training, are generally geared towards forward-looking predictive tasks which may echo theoretical aspects of short-term planning found in human cognition. However, their potential limitations in handling reciprocal context relationships could reflect a smaller scope in understanding the complex, interconnected structure of discussions typically navigated by the prefrontal cortex.

ELECTRA-related Models (DeBERTa V3): Models such as DeBERTa V3, which benefits from ELECTRA's pre-training approach for distinguishing subtle textual discrepancies, could be expected to perform comprehensively in tasks that require a deep dive into the content. This might parallel the human capacity for Theory of Mind Encoding and Emotional and Social Cognition, allowing for a nuanced perception of the deliberative nature of discussions and the emotional currents within them.

5.7.2 Expectations for Specific LLMs:

GPT-2: Given GPT-2's generative prowess, it may favor tasks aligned with linear information processing, resembling aspects of narrative cognition. Its limitation in bidirectional understanding may be counterbalanced by its capability for sequential content rendering, which could, in some respects, mirror human strategies for constructing and following narratives.

DeBERTa v3: Known for its Gradient Disentangled Embedding Sharing, DeBERTa v3 aims to spatially disentangle the embeddings it uses, potentially offering robustness in tasks requiring discernment of complex discussion contours.

BART: BART's competencies suggest an affinity for tasks that demand both synthesis and generation of textual information. It may replicate the iterative contextual understanding and

expressive adaptability, akin to human processes of self-evaluation and communicative adjustment, where discourse is reshaped as it unfolds.

ALBERT: As ALBERT employs parameter-reduction techniques, it may particularly excel in scenarios where processing efficiency is paramount, potentially aligning with human fast-paced decision-making processes. However, its capacity to handle complex, multi-layered discussions may be limited as a trade-off for efficiency.

Overall, the distinct architectural and training attributes of each LLM inform the projected expectations. BERT-related models may display strengths in bidirectional and contextual nuances, DeBERTa v3 is anticipated to excel in discerning intricate textual patterns, GPT-2 could demonstrate facility in generative tasks, and ALBERT might show competence in tasks where processing efficiency is crucial. These prognostications will be integral to the evaluation of LLMs' capacity to approximate various neurocognitive functions involved in discussion comprehension.

6 Evaluation and Validation

Before aggregating results and taking averages etc. the type of distribution of the results is analysed such that that can be taken into account. Applying the Shapiro-Wilk Test to determine the p-value and filtering for p-value > 0.05 (⇔normal distribution) this table shows the LLMs and the respective task where the fold, seed pairs are normally distributed (11 out of 60). So mainly the data is not normally distributed which means we have to take that into consideration for deciding an apt statistical method.

```
shapiro_results[shapiro_results["p-value"] > 0.05]
```

Executed at 2023.12.31 00:57:21 in 338ms

11 rows 11 rows × 5 columns pd.DataFrame ↗

	project_name	model	Condition	Shapiro-Wilk	p-value
6	1 Stance Alignment Task - No Context	bert-base-uncased	No Context	0.915260	0.052778
15	2 Sequential Coherence Task	albert-base-v2	Context	0.923070	0.113537
16	2 Sequential Coherence Task	bert-base-uncased	Context	0.925346	0.125606
18	2 Sequential Coherence Task	gpt2	Context	0.909117	0.061316
19	2 Sequential Coherence Task	microsoft/deberta-v3-base	Context	0.946679	0.319476
21	3 Interactive Dynamics Task	bert-base-uncased	Context	0.981560	0.096600
42	5 Discussion Contour Recognition Task	facebook/bart-base	Context	0.966898	0.688497
44	5 Discussion Contour Recognition Task	microsoft/deberta-v3-base	Context	0.941881	0.260172
45	5 Discussion Contour Recognition Task - No Context	albert-base-v2	No Context	0.959224	0.295911
46	5 Discussion Contour Recognition Task - No Context	bert-base-uncased	No Context	0.935890	0.118926
47	5 Discussion Contour Recognition Task - No Context	facebook/bart-base	No Context	0.948388	0.153032

Figure 44 - Shapiro-Wilk Test on probing results

6.1 Does the context influence the Performance significantly across all Probing Tasks?

The probing framework allows one to easily append or not append the root claim (topic /context) to the input. So runs across probes 1 to 5 (excluding the simplified versions) for context and non-context versions were logged and are evaluated in this Chapter. First, to investigate the influence of appending the context topic (the root claim) to the input claims on the performance of the LLMs for probing tasks 1 to 5 (unfortunately, the light tasks crashed again; however, it is mainly about the relative differences) the distribution in boxplots context and no-context tasks was

analysed to find out if all unique seed, fold pairs should be averaged or the median be taken to compare the LLMs over the projects.

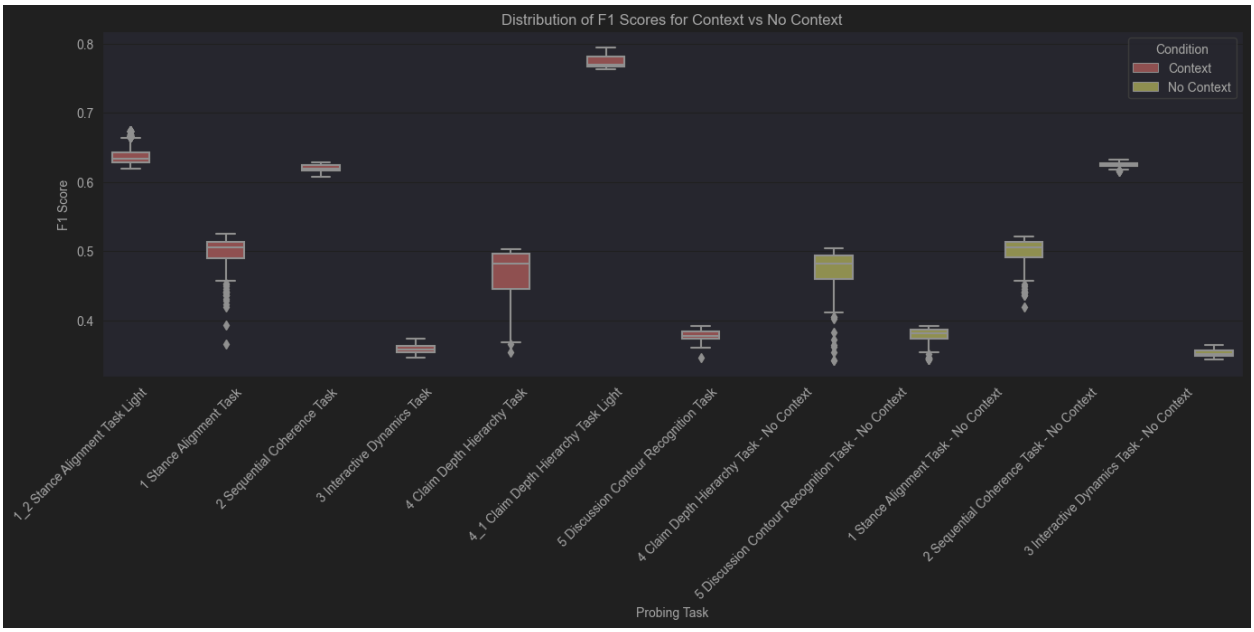

Figure 45 - Probing Task Results Distribution

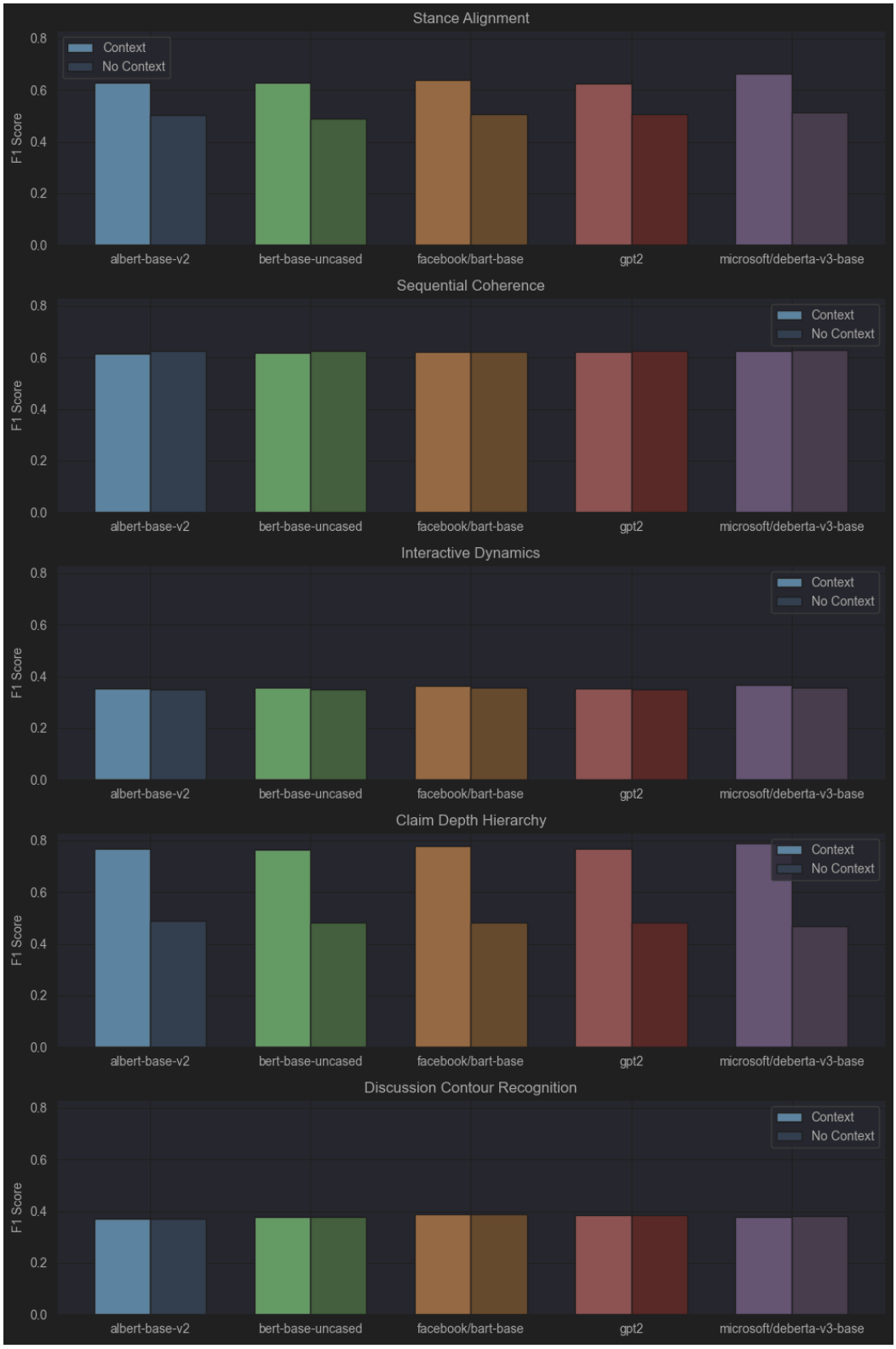

Figure 46 - Context Topic / No Context Topic

A clear difference is observable for Stance Alignment: Appending the context topic (root claim) to the input claim helps the model solve the task. This seems plausible since the labels were calculated concerning the root claim first (multiplication of all stances from the root claim until the selected claim).

Also, the claim depth hierarchy is positively influenced by the topic. It also seems plausible since to understand a hierarchy, one needs to know the ground to understand the relative differences to this reference point.

Only immediate claims are considered for recognizing sequential coherence; the topic does not make a significant difference.

For Interactive Dynamics, I had expected that the context would make a difference since if a claim is said on some topic, that claim could cause a different number of reactions than in other topics (imagine a political debate where some opinions are more biased in one group than in another) However it seems that this outweighed by more recent claims/context.

For Discussion Contour Recognition, the observations again make sense since they are about the potential claim and depth space between two claims, mainly influenced by those claims straining the contour in between.

6.1.1 No-Context Ranking

The performances across none, randomization, and permutation tasks were grouped by LLM and probing task and then aggregated by the median. The differences between the probing task and the no-context performances were calculated and normalized by the overall sum of those 2

performances. After that, all the differences were again averaged and finally ranked from 1 to 5.

```
category_rankings[["index","no_context"]].sort_values(by=['no_context'])
Executed at 2024.01.02 03:40:54 in 401ms
```

5 rows × 2 columns pd.DataFrame

	index	no_context
0	albert-base-v2	1.0
3	gpt2	2.0
1	bert-base-uncased	3.0
4	microsoft/deberta-v3-base	4.0
2	facebook/bart-base	5.0

Figure 47 - No-Context Ranking

In evaluating the impact of contextual information on language model performance, a comparison was conducted using selectivity differences as the criterion. ALBERT-base-v2, characterized by its parameter-sharing and embedding factorization techniques, exhibited the smallest decline in performance when the context was absent. This seems consistent with the selectivity findings where simplifying tasks favoured ALBERT's performance exceptionally well.

GPT-2 is also expected to do better in tasks that are not heavily weighted on complex context knowledge or long-range dependencies where BERT, BART, and DeBERTa are more favored. BERT-base-uncased, positioned third, indicates that its bidirectional context processing and deep language structure understanding partially mitigate the effects of context removal. This observation is consistent with its architecture that emphasizes contextual relationships, albeit with a discernible impact in context-absent scenarios.

DeBERTa-v3-base's performance was notably affected, suggesting that its training methodology, which heavily emphasizes context for predictions, may not be as effective when such context is stripped away. This was unexpected given DeBERTa's advanced pre-training techniques.

Facebook/BART-base demonstrated a significant reliance on context, as evidenced by its last-place ranking. This outcome is indicative of its seq2seq architecture's design, which is optimized for text generation tasks where context is integral.

The models' collective performance suggests that the impact of context removal is not uniformly distributed across different architecture types or model groups. While one might expect the BERT and ELECTRA groups, known for their contextual processing strengths, to excel, the results indicate that specific architectural and training features may be more crucial role in determining model robustness in no-context conditions. This assessment underscores the subtle interplay between model design, training data, and task-specific demands rather than a straightforward classification based on model groups.

6.2 Control Tasks

This subchapter explores the differences between the linguistic probing tasks (“none” control task) and the control tasks “randomization” and “permutation.”

6.2.1 Selectivity Comparison

To get an overview of LLMs performances, the U-tests were conducted to assess model performance differences between the linguistic tasks and control tasks across folds and seeds. The dataset grouped by individual fold seed pairs lies in a a range of observations per group: minimum 12 and maximum 42. This indicates a sufficient number of data points for this distribution comparison:

```
observations_per_group = wandb_df.groupby(['model', 'project', 'seed', 'fold']).size()


min_observations = observations_per_group.min()
max_observations = observations_per_group.max()

min_observations, max_observations
Executed at 2023.12.31 22:43:29 in 835ms

(12, 42)
```

Figure 48 - Folds, Seeds Count Range

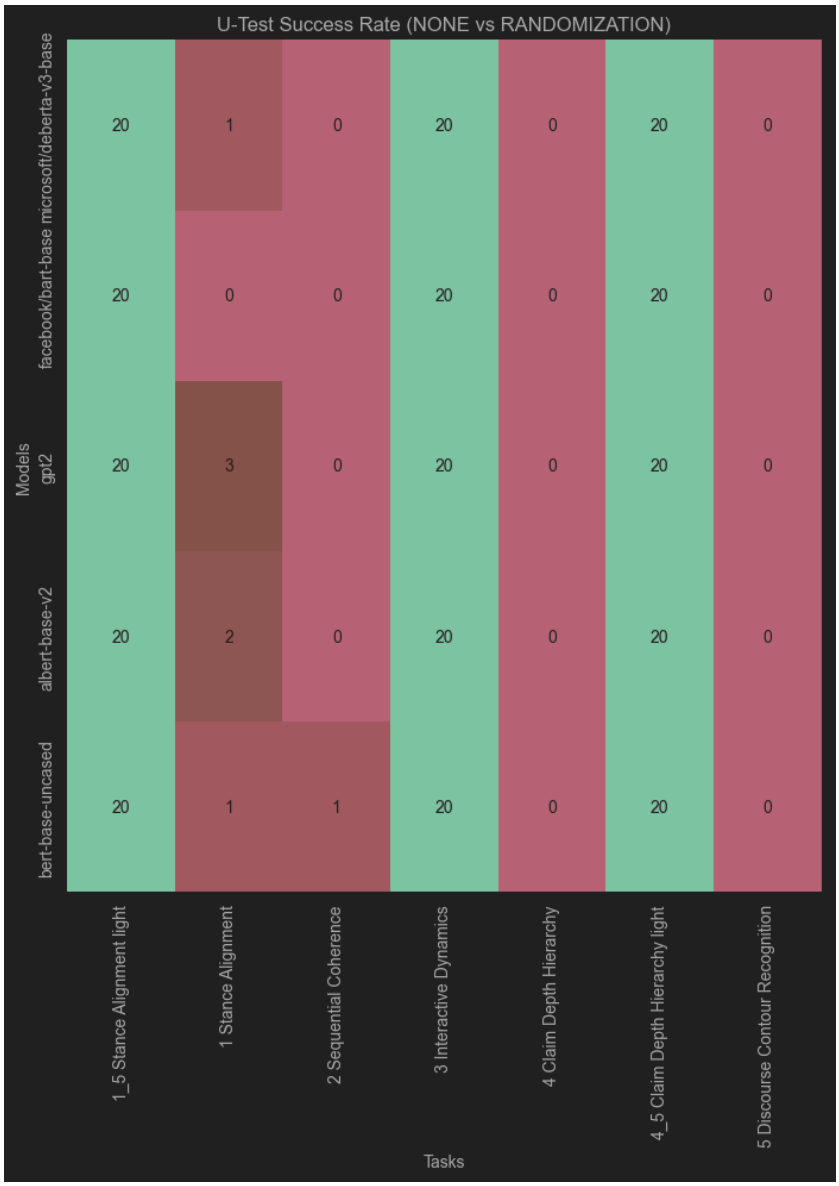

Figure 49 U-Test : None vs Random

The LLMs seem to do well in the simplified tasks (1_5, 4_5) , task 3 and a few runs in task 1 but otherwise show no significant difference in the fold, seed distribution using the U-test for the indicating that those tasks were indeed too complex to solve and would require further redesign exceeding the time frame of this thesis.

The alpha level for these tests was set at 0.05, aligning with standard scientific practice. This level is widely accepted for determining statistical significance, indicating a 95% confidence in the test results. The choice of this threshold, while conventional, carries inherent limitations. Moreover, the U-test does not measure the magnitude of differences, only their statistical significance. Therefore, while the U-test can confirm whether model performance statistically differs from randomization, it does not quantify the superiority or provide detailed insights into the degree of that difference, which allows for model comparison across all tasks.

To address this limitation and offer a more granular analysis of model performance, this methodology is expanded to include selectivity ranking. Selectivity is calculated as the median difference in performance between the linguistic tasks (NONE) and the randomized control tasks (RANDOMIZATION), aggregated across all seeds and folds for each model and task. This method moves beyond the binary outcome of the U-test—significant or not significant—to measure how much more effectively a model performs compared to a random baseline when considering the overall tendency of its performance across various initializations and data splits. It allows for a deeper comparison between models, highlighting not only whether they can outperform randomness, as indicated by the U-tests, but also the extent to which they do so, as reflected in the selectivity scores**.** A high selectivity indicates that a model can discern and leverage the linguistic structure within the data, thereby efficiently distinguishing between related and unrelated content in the task. Low or negative selectivity points to a model's struggle

to surpass a random guessing strategy, which could be due to various factors such as model architecture limitations, misalignment with training data, or the inherent difficulty of the task itself.

So for the ranking the following boxplots are observed:

Probing Task 1 / 1_5: Stance Alignment

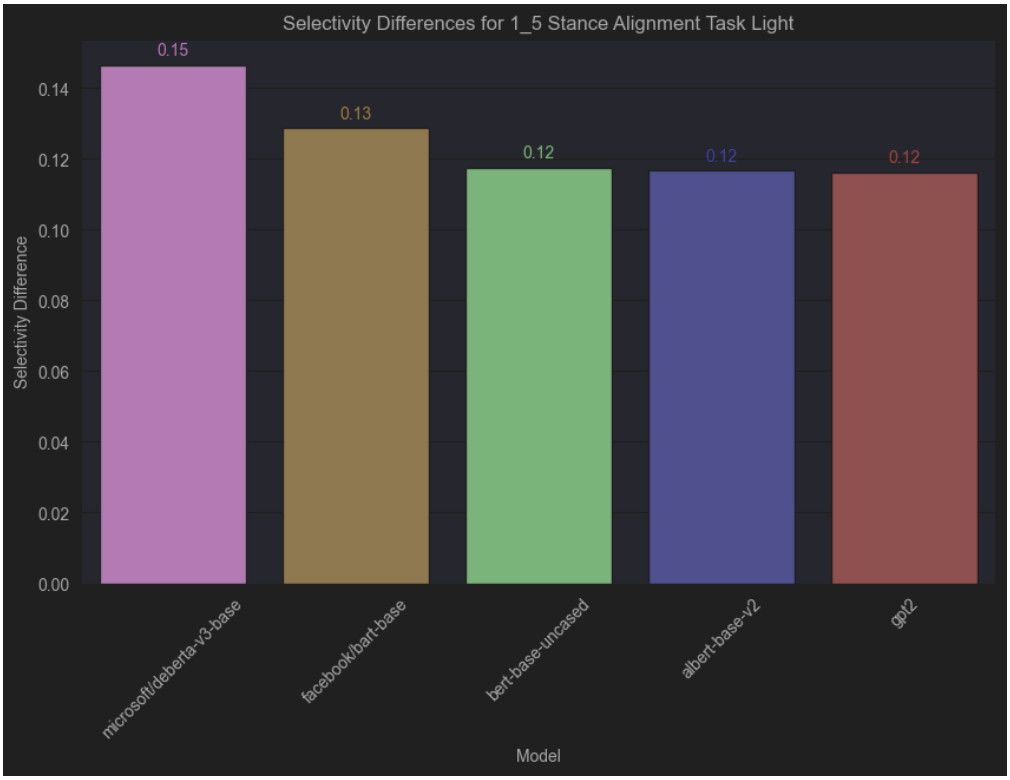

Figure 50 - Evaluation: Probing Task 1_5

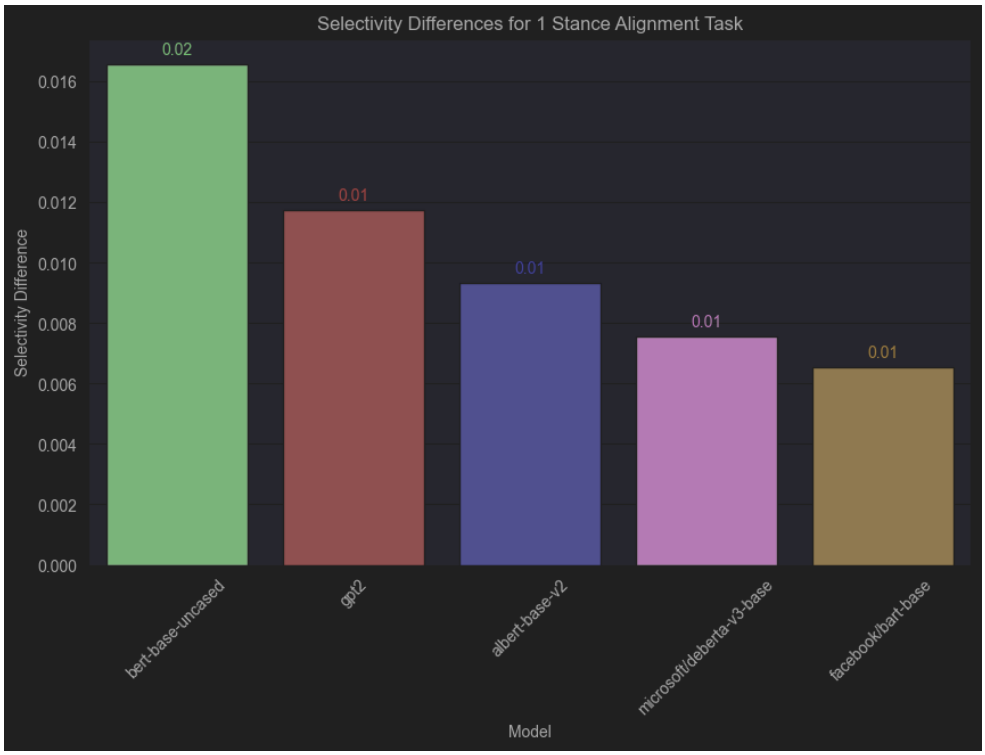

Figure 51 - Evaluation: Probing Task 1

1. **BERT (bert-base-uncased)**: This model performs well in tasks requiring deep contextual understanding due to its bidirectional encoder and learning specific patterns needed for tasks such as sentiment analysis. However, it is interesting to note that its selectivity is lower for Task 1 light than Task 1. This could be due to the simplified nature of Task 1 light, which might leverage BERT's strength in processing complex relational structures less than Task 1 does.
2. **GPT-2 :** Known for its predictive capabilities in generating text based on preceding context, GPT-2's architecture may need to be better-suited for bidirectional stance alignment than BERT's. This might explain its lower selectivity in Task 1. However, its performance improves in Task 1 light due to the reduced complexity of the task aligns better with its unidirectional processing capabilities. Still, other models favour more from

the reduced complexity, which makes it look relatively weak in the ranking for task 1 light.

3. **BART (facebook/bart-base)**: The lower selectivity in Task 1 Light might be attributed to the task's simplified nature. BERT's capabilities are possibly underutilized here compared to Task 1, which likely involves more complex relational structures that BERT is adept at processing.
4. **ALBERT** (albert-base-v2): ALBERT's parameter efficiency and reduction techniques may limit its depth of contextual understanding, impacting its selectivity for these two tasks. Due to its compressed nature (compared to its bigger relative BERT, which has many more parameters) it slightly improves in the simplified Task 1 light where input claims are closer related.
5. **DeBERTa** (microsoft/deberta-v3-base): With its innovative architecture and ELECTRA-style pre-training, DeBERTa is expected to perform well across a broad range of NLU tasks. Still, its general strength may not include sentiment analysis much, which requires a specific pattern that favours overfitted / specific LLMs more than generalists. Task 1 light shows the best selectivity, indicating that the simplified nature of the task may better align with its generalization strengths.

Probing Task 2: Sequential Coherence

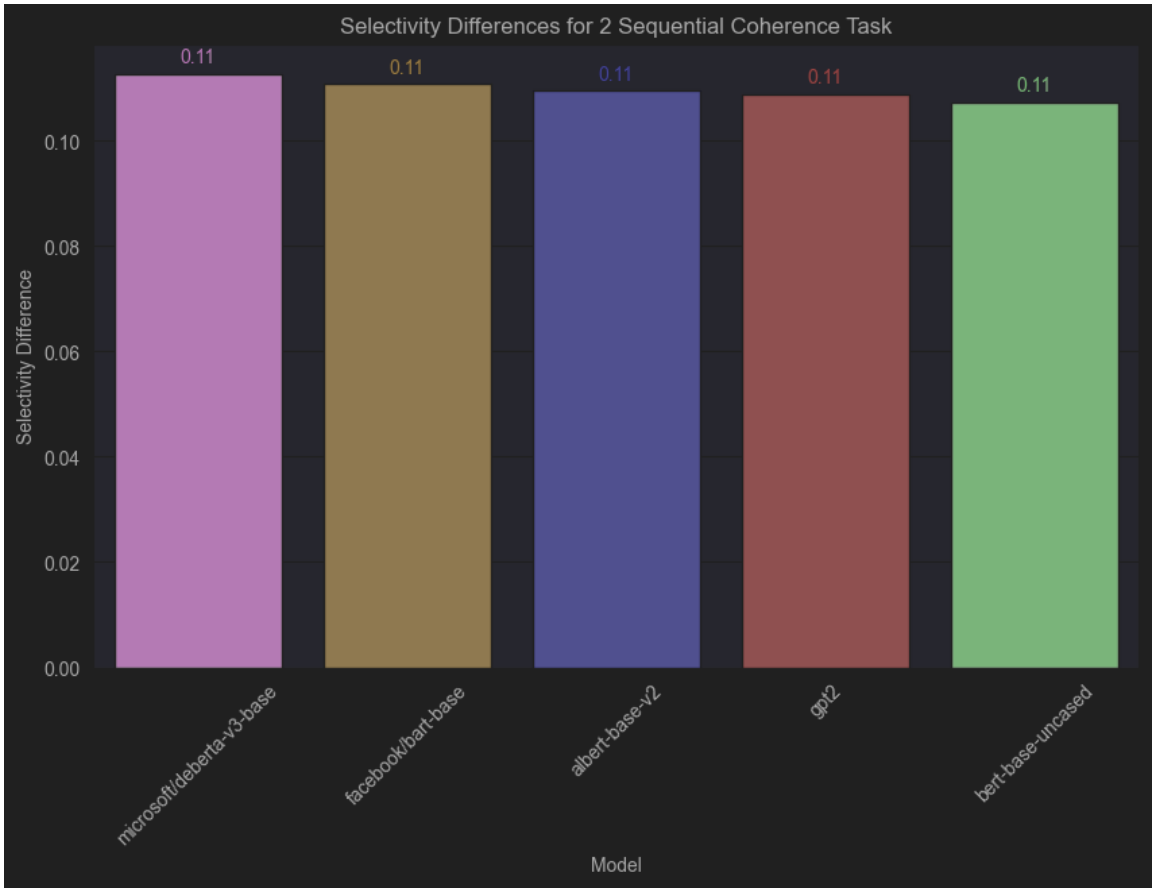

Figure 52 - - Evaluation: Probing Task 2

This task evaluates whether two claims are immediately sequential in a discussion. It requires understanding the flow and structure of the conversation. The LLMs perform similarly poorly in this task, as indicated in the U tests. But why? Upon reinspection of the data for the task it became evident what probably is the cause of this overall bad performance: the website data cleaning has been forgotten. So, the LLMs had to solve the tasks with a lot of noise in the data which could explain those performances.

1. **DeBERTa (microsoft/deberta-v3-base)**: Its advanced architecture with its pretraining on a large corpus, is designed to provide a better understanding of the context, which can help handle noise.

2. **BART (facebook/bart-base)**: Used denoising pretraining, which is made to filter noisy data.
3. **ALBERT´s** Sentence ordering prediction (SOP pre-training is beneficial for the task of finding if the claims are in (direct) order or not. Its smaller size and efficiency help it to deal with the noise compared to its bigger brother BERT, which may “see too much” in the noise and get confused.
4. **BERT (bert-base-uncased)**: Similar to ALBERT, BERTs next sentence prediction (NSP) pretraining task should give it a clear advantage. As mentioned before, it´s probably too slow or too deep to handle the noise as efficiently as ALBERT.
5. **GPT-2 (gpt2)**: Despite being second in ranking, GPT-2's lower performance relative to BERT could be due to its unidirectional nature, focusing primarily on the preceding context. This can be a disadvantage when identifying whether a claim is an immediate response to another.

Probing Task 3: Interactive Dynamics

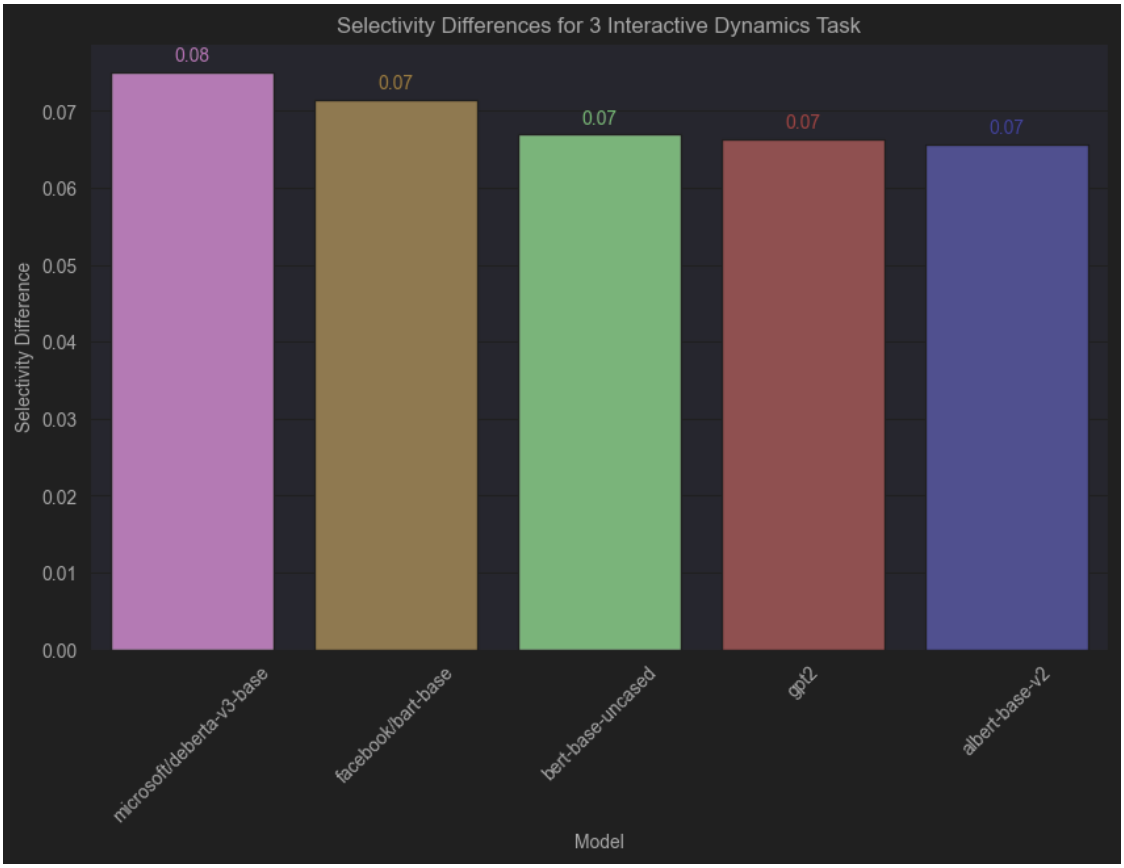

Figure 53 - Evaluation: Probing Task 3

This task predicts the potential engagement a claim might incite, categorized by the number of subsequent related claims. The distribution table of this task in Chapter 5.9.10 shows the most significant number of unique topics. So LLMs with broad contextual understanding have a benefit here.

1. **DeBERTa (microsoft/deberta-v3-base)**: It leads the ranking, suggesting its architecture and training are well-suited for tasks involving the prediction of subsequent discussion dynamics, likely benefiting from its Gradient Disentangled Embedding Sharing technique which helps in disentangling the positions of words from their word embeddings. This means that the model can better understand the individual contribution of each word in a sentence, apart from where it is placed in the sequence. It may use the stance/sentiment of

the sentence and its general advanced contextual comprehension to estimate whether the claim might cause a lot of responses (provocation, surprise). Together with its broad understanding it´s a winner.

2. **BART (facebook/bart-base) vs BERT and ALBERT**: Its second-place ranking could be due to its seq2seq structure. This allows BART to learn how the current claim influences the next claim, and how the next claim influences the following claim, and so on. This contrasts with models like BERT and ALBERT, trained on Masked Language Modeling (MLM) tasks. MLM tasks involve predicting the masked word in a sentence, given the other words that are not focused on looking ahead as much as seq2seq.
3. **BERT vs ALBERT**: Understanding the context well and giving an appropriate response favours models with broader contextual understanding. So, this time BERT is ranked before his brother.
4. **GPT-2**: Because of its generative and more creative nature and unidirectional encoders, it is less apt to look from both sides (e.g., What is the most appropriate thing to say in this situation AND what could be the consequences/reactions? Is it (then) still the most appropriate thing to say?).

Claim Depth Hierarchy (Probe 4)

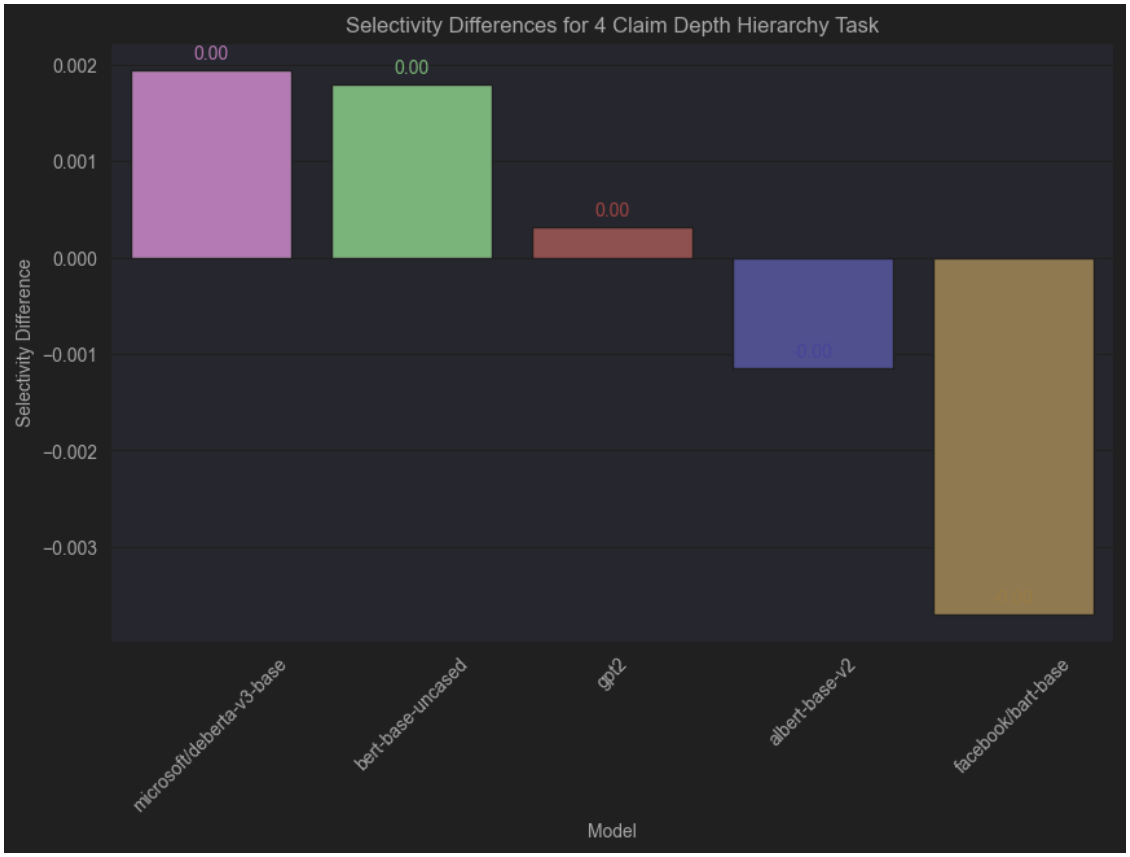

Figure 54 - Evaluation: Probing Task 5

1. **DeBERTa** This model's architecture allows for a deep understanding of the context, and its training on a diverse dataset could give it an edge in recognizing complex patterns like hierarchical relationships. DeBERTa's performance may reflect its advanced attention mechanism that can more effectively capture the relationships and hierarchies between different text parts.
2. BERT's bidirectional context understanding could be beneficial in recognizing hierarchical relationships, but it may not be as effective as DeBERTa, which is optimized further for such tasks.
3. **GPT-2's** strength lies in generating coherent text sequences. While it's not specifically designed for hierarchical tasks, its large-scale training on diverse text may have

incidentally provided it with some understanding of hierarchical structures in language. That said, GPT-2 might struggle when the hierarchy gets more complex and node distances increase (not favoring its generative nature).

4. **ALBERT's** parameter-sharing and sentence-ordering prediction pre-training tasks may not be as practical for complex hierarchical understanding.
5. **BART's** sequence-to-sequence model is geared towards text generation tasks and may not be as adept at recognizing complex/subtle hierarchical relationships.

Claim Depth Hierarchy Light (Probe 4.5)

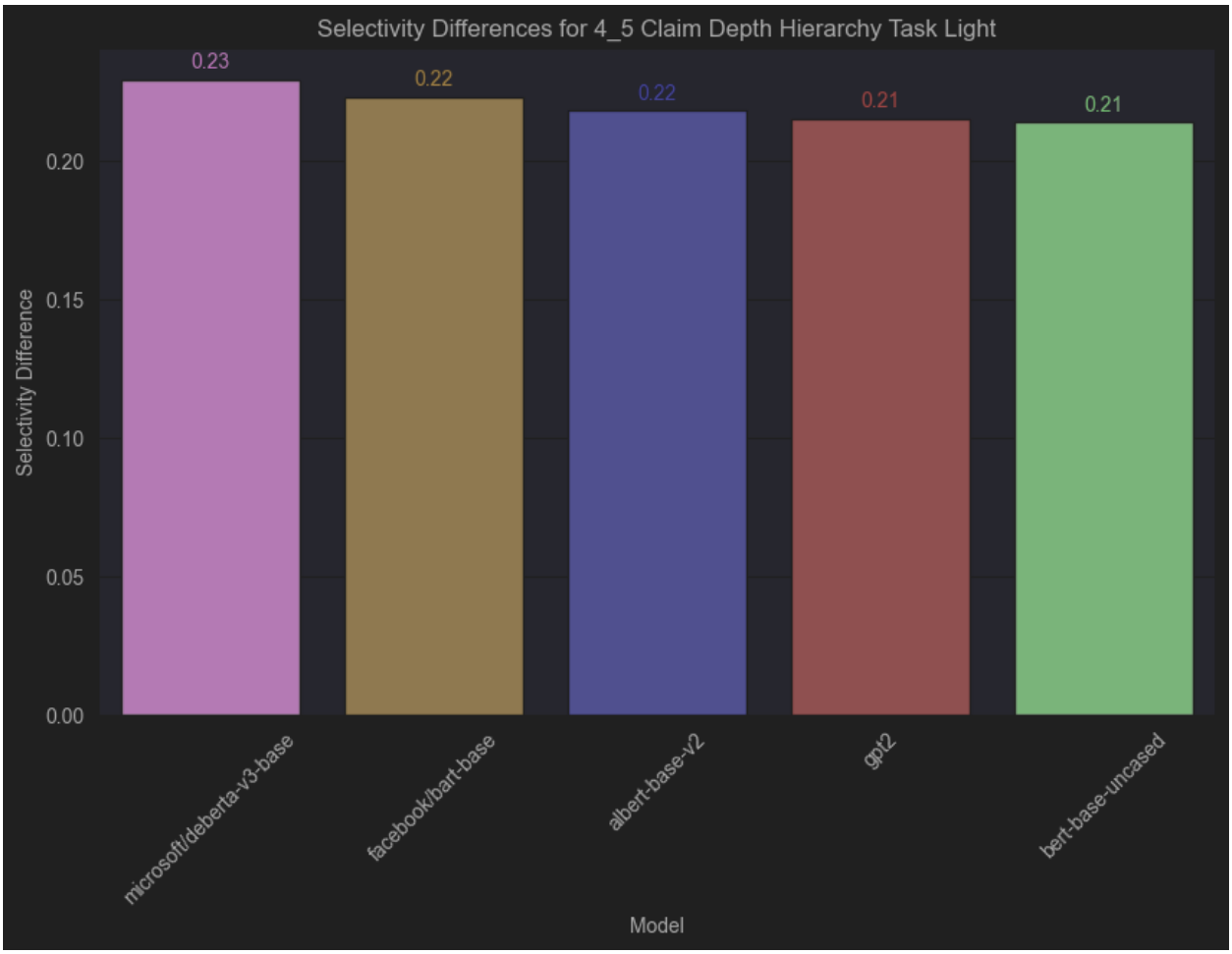

Figure 55 - Evaluation: Probing Task 4_5

In the simplified task, all models show positive selectivity, with **DeBERTa** and **BART** leading. The simplification likely makes the hierarchical relationships more direct and more accessible to detect. It again allows DeBERTA to use its well-developed general contextual understanding.

1. **DeBERTa v3** stays ahead in this task.
2. **BART** : Despite their low performance on the more challenging version of this task, the constraint version, which brings the input claims closer together, is more in line with its generative, continuous nature.
3. The simplified version again favors ALBERTS' efficient and compressed nature and does not demand a deep understanding of version 4.
4. BERT shows the opposite pattern of ALBERT again: While simplified task 4_5 may not favor his deep understanding as much as ALBERTS compression, his contextual understanding benefits him in task 4.
5. **GPT-2** again benefits from the simplification, similarly as in task 1.

Discussion Contour Recognition (Probe 5)

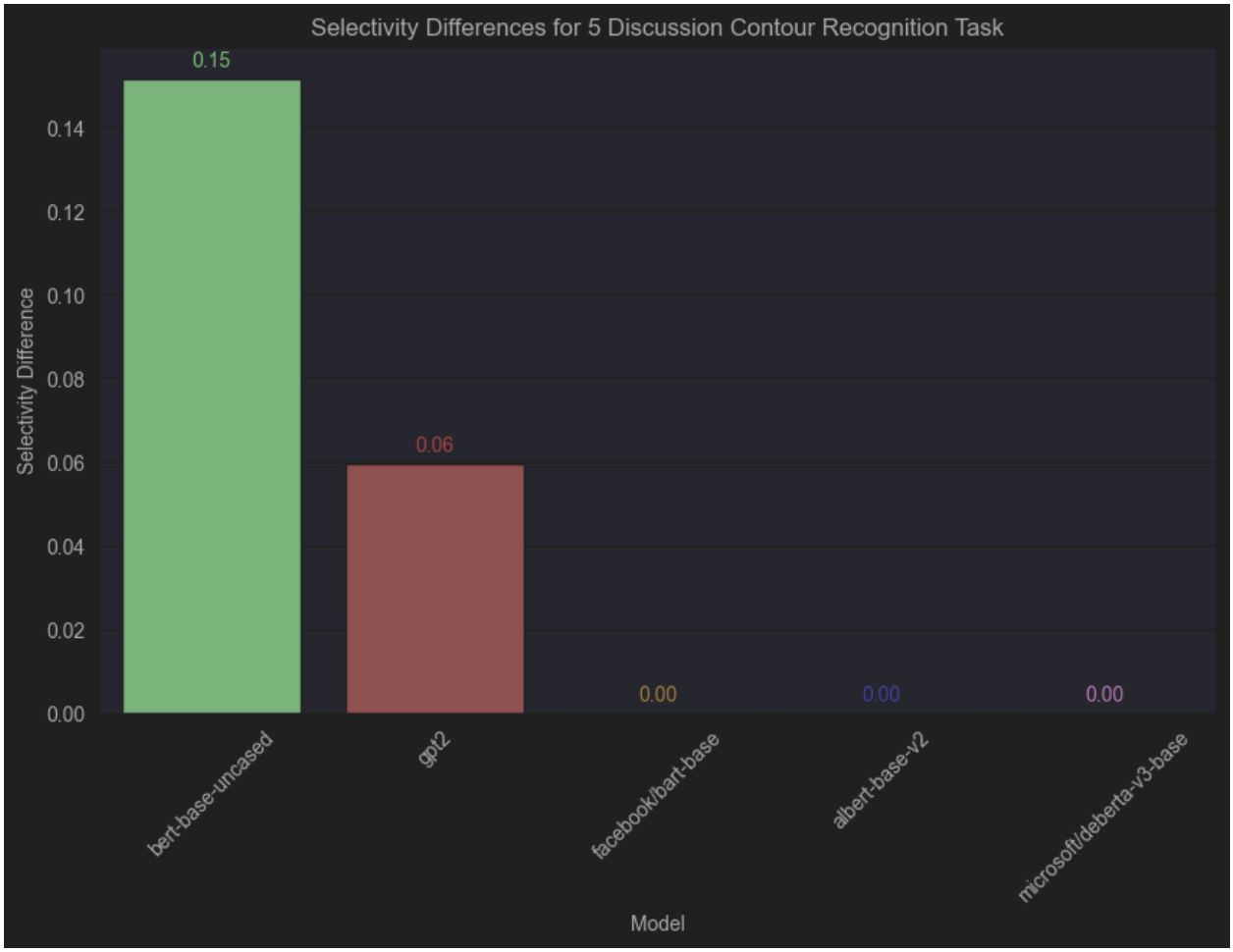

Figure 56 - Evaluation: Probing Task 5

BERT 's architecture, which is designed to understand context within and between sentences, may give it an advantage in estimating the number of claims between two points. The model's bidirectional context understanding could help it anticipate the progression of discussion, leading to its first place in this task.

GPT-2 is known for its ability to generate coherent text sequences, and this skill might translate well into predicting the flow and density of discussions. Still, if the claims are too far from each other, it may struggle more.

BART's encoder-decoder structure may not be as efficient as BERT or GPT-2 in predicting discussion contours. Since this task entails, task 4 as well (namely the number of depth layers traveled / hierarchical thinking), where BART could have performed better, the results here are less of a surprise.

ALBERT again needs features/parameters for this rather complex task.

While generally leading the charts, **DeBERTa** does surprisingly bad in this task. It could be a similar cause as in task 1: Some specialist' specific patterns do not suit the generalist.

6.2.2 Selectivity Ranking

By aggregating the ranking results from all 7 probing tasks and averaging the final probing task ranking is received :

```
category_rankings[["index","selectivity"]].sort_values(by=['selectivity'])
```

Executed at 2024.01.02 03:39:51 in 369ms

5 rows 5 rows × 2 columns pd.DataFrame

	index	selectivity
4	microsoft/deberta-v3-base	1.0
1	bert-base-uncased	2.0
2	facebook/bart-base	3.0
3	gpt2	4.0
0	albert-base-v2	5.0

Figure 57- Overall Selectivity Ranking

These findings verify the findings mentioned in Section 2.3.3 ::Potentially BART had more capacity but still is outperformed by BERT confirming findings of Section 2.3.1. Similarly for DeBERTa , who leads again and confirms the findings from Section 2.4.1.

```
category_rankings[["index","permutation"]].sort_values(by=['permutation'])
```

Executed at 2024.01.02 03:40:26 in 359ms

5 rows × 2 columns pd.DataFrame

	index	permutation
4	microsoft/deberta-v3-base	1.0
0	albert-base-v2	2.0
1	bert-base-uncased	3.0
3	gpt2	4.0
2	facebook/bart-base	5.0

Figure 58 - Permutation Ranking

Some trends continue: BERT outperforms BART and DeBERTa leads

DeBERTa v3, with its disentangled attention mechanism, is anticipated to be less reliant on token position, favoring robustness in the face of permuted inputs. The ranking confirms this expectation, placing DeBERTa V3 at the top, indicative of its enhanced capacity to process language beyond token-level dependencies.

ALBERT's ranking as second might surprise, given its design for parameter efficiency, which could suggest a more limited understanding compared to larger models. However, its performance points to effectively learning sentence structures transcending token positions. Its masked language modeling (MLM) objective and sentence-order prediction (SOP) pre-training task could influence it. The MLM task, where random words are masked, and the model predicts them, could contribute to ALBERT's ability to understand language at the token level, as it forces the model to learn the context of individual words and their probable substitutes. This could enhance its robustness against permutations since the model would rely less reliant on the

specific ordering of tokens to make predictions. The SOP task, which involves determining whether two sentences follow each other in the correct order, also plays a role. While SOP is more focused on sentence-level understanding, it requires the model to grasp the broader narrative and coherence within text passages, which could help in tasks where the input is somewhat disordered.

BERT, positioned third in the ranking, aligns with expectations, considering its bidirectional training and reliance on positional encodings. Its middle-of-the-pack performance suggests a balanced sensitivity to token position, reflecting its design for contextual understanding within sentences.

GPT-2, ranking fourth, reveals a more significant impact from permutation, likely due to its unidirectional, autoregressive training. Despite its large parameter count and extensive training data, the model's design for next-token prediction implies a dependency on token order, affecting its robustness when inputs are shuffled.

While **BART**'s seq2seq architecture and exposure to noised text during training might suggest adaptability to disordered input, its core strengths lie in text generation and comprehension within structured contexts. Its ranking reflects a nuanced capability: adeptness in handling tasks with coherent sequences but less resilience to heavily permuted structures. It aligns with its design as an encoder-decoder model that may rely on sequence integrity for optimal performance.

In examining the trends among model groups about their robustness against input permutation, the ELECTRA group, exemplified by DeBERTa, exhibits considerable resilience. This robustness can be attributed to the ELECTRA group's innovative attention mechanisms and

pre-training strategies, which may endow them with enhanced adaptability to perturbations in input data.

On the other hand, the BERT group, which relies on bidirectional encoders, displays varying degrees of robustness to permuted inputs. This variation indicates that factors beyond model size and the diversity of training data contribute to a model's robustness. The bidirectional nature of BERT's encoder allows it to understand context from both directions within a sentence hence it may be more sensitive to the order of words. When inputs are permuted, the disruption of the learned linguistic patterns could affect the model's performance.

U Test Results

Those show a one-sided Whitney Mann U Test assessing if the permutation distribution significantly different from the permutation distribution using a significance score of 0.05. Only a few significant results using this method are visible here. However, this thesis focuses on relative comparisons of the models, which is better captured in the differences explored in the last section.

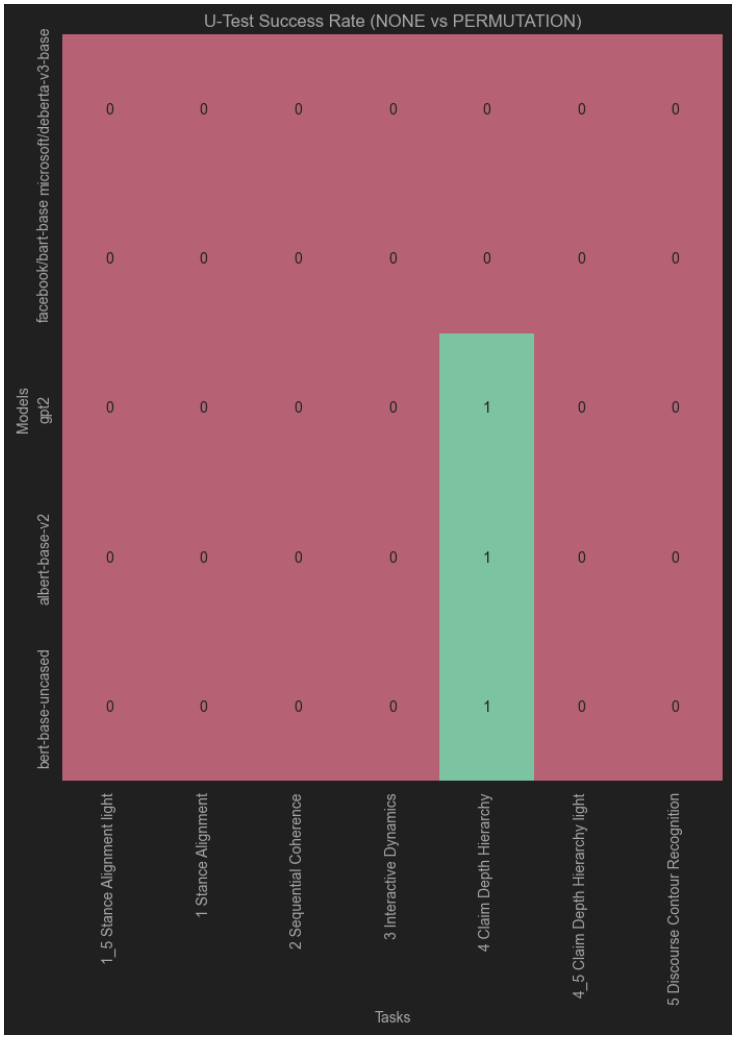

Figure 59 - UTEST NONE vs PERMUTATION

6.3 Result Variations across different folds or seeds

The implementation of ANOVA involved two distinct analyses: one focusing on seed variability and the other on fold variability. In the seed variability analysis, the data folds were held constant while the seeds were varied. This allowed for the assessment of how fluctuations in the initialization parameters (seeds) impacted the performance of LLMs.

Conversely, the fold variability analysis kept the seeds constant and varied the data folds, thus gauging the influence of different data splits on model performance.

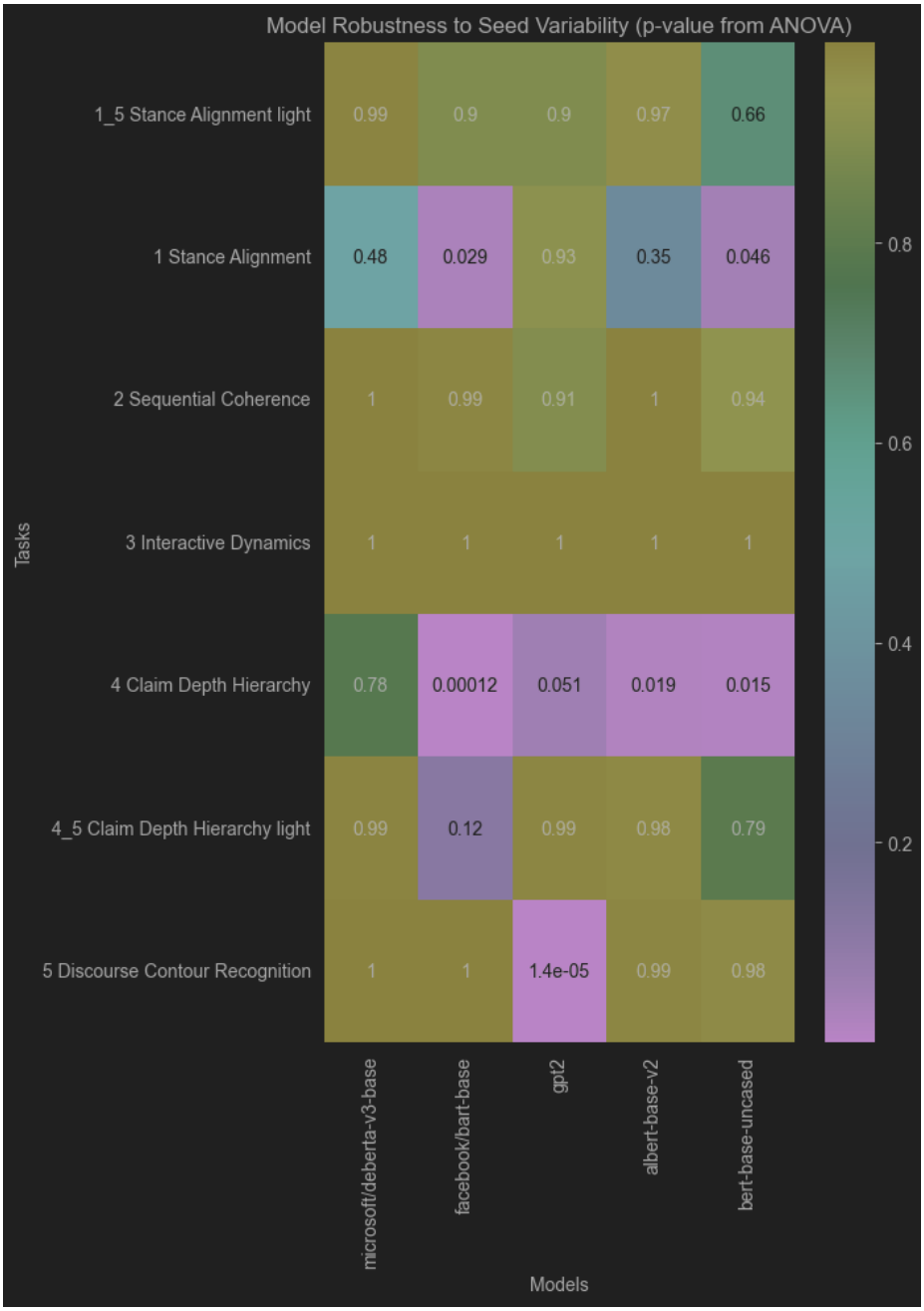

Figure 60 - Seeds ANOVA results

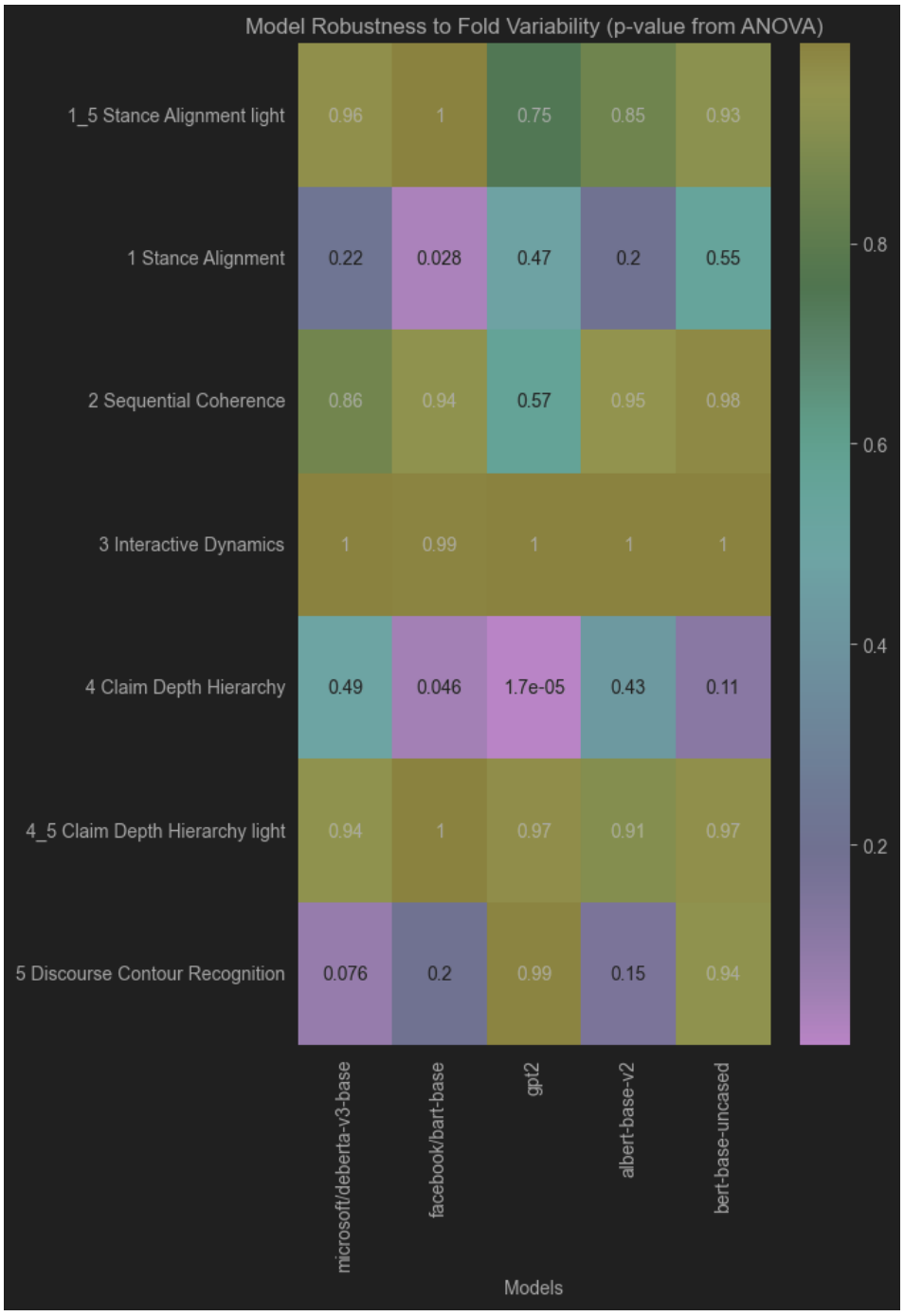

Figure 61 - Folds ANOVA results

The analysis of seed variability revealed distinct patterns among different language models. Microsoft's **DeBERTa,** for instance, demonstrated remarkable robustness to changes in seed initialization. This stability can be attributed to its advanced architecture, which includes

Gradient Disentangled Embedding Sharing, enhancing its ability to maintain consistent performance regardless of initial conditions.

In contrast**, BERT**, specifically the bert-base-uncased model, exhibited more variability in response to seed changes. This variation suggests that despite BERT's bidirectional encoding capabilities, which provide a comprehensive understanding of context, its performance can be more sensitive to the randomness introduced by different seed initializations. This sensitivity might be reflected in BERT's reliance on contextual relations within the text, which can be influenced by variations in the initial training setup.

Regarding fold variability, the analysis again highlighted different responses among the LLMs. Task 4 and Task 1, known for their high context dependency, caused the most fluctuations in model performance. This outcome aligns with the expectation that models trained on diverse web-scraped data, which can be biased towards specific topics, would exhibit variability in performance when exposed to different data splits. Such variability underscores the models' reliance on the particular nature of their training data (also mentioned in *UL2: Unifying Language Learning Paradigms* by (Tay et al. , 2022) and their ability to generalize across varied contexts.

Furthermore, the fold variability analysis highlighted how models like bert-base-uncased managed to maintain a more stable performance across different data folds compared to others. This stability could result from BERT's architecture, adept at synthesizing contextual information and may allow for more consistent performance even when the data split varies.

6.3.1 Robustness Ranking

```
category_rankings[["index","robustness"]].sort_values(by=['robustness'])
Executed at 2024.01.02 03:41:31 in 1s 170ms
```

5 rows × 2 columns pd.DataFrame

	index	robustness
4	microsoft/deberta-v3-base	1.0
1	bert-base-uncased	2.0
3	gpt2	3.0
0	albert-base-v2	4.0
2	facebook/bart-base	5.0

Figure 62 - Robustness Ranking

DeBERTa's top placement in robustness, as indicated by the ranking, is consistent with its design attributes, particularly the Gradient Disentangled Embedding Sharing. This feature likely contributes to its resilience against the variability introduced by different seed settings during initialization. DeBERTa can maintain stable performance regardless of the seed variations by effectively disentangling the gradients for different types of embeddings, highlighting its strong generalization capabilities.

BERT's performance shows moderate variability, occupying the middle ground in the ranking. This outcome could stem from the model's bidirectional nature, which, while robust in capturing contextual dependencies, might exhibit sensitivity to the randomness of seed initialization. Such sensitivity suggests that BERT's performance is closely tied to the specificities of its training setup, leading to more pronounced effects when the initial conditions change.

GPT-2 and **ALBERT**, falling below DeBERTa and BERT, may experience a more significant impact from seed and fold variability due to their respective architectures and pre-training methodologies. GPT-2's autoregressive nature and ALBERT's parameter-efficient design might not afford the same level of stability when confronted with variations in the data.

Fold variability, which assesses model consistency across different data splits, further emphasizes the importance of training data in model robustness. The models' performance on probing tasks, particularly those with high context-dependency like Task 4 and Task 5, revealed fluctuations that could be indicative of how well these models generalize from their training data to varied contexts. Models that exhibit less fluctuation across folds suggest a more vital ability to generalize and are less dependent on the specific distribution of their training data.

In summary, the robustness ranking, as seen in the image, underscores the significance of architectural design and training data diversity in determining a model's robustness. DeBERTa's architecture equips it well for consistent performance across seed and fold variations. In contrast, BERT's variability suggests that its performance is influenced by the randomness of initialization and data splits.

6.4 Evaluation Summaries

The following table shows the different performances in linguistic probing tasks, randomization and permutation grouped by MODEL,PROBE and aggregated by median. The selectivity here is not normalized as it is has been for comparison of LLMs.

'MODEL'	'PROBE'	'NONE'	'PERMUTATION'	'RANDOMIZATION'	'SELECTIVITY'
albert-base-v2	1 Stance Alignment	0.503131	0.504879	0.486931	0.016200
""	1_5 Stance Alignment light	0.628084	0.594903	0.485260	0.142824
""	2 Sequential Coherence	0.616286	0.603290	0.485122	0.131164
""	3 Interactive Dynamics	0.354439	0.341555	0.256809	0.097630
""	4 Claim Depth Hierarchy	0.484198	0.476132	0.493685	-0.009487
""	4_5 Claim Depth Hierarchy light	0.768255	0.739047	0.484141	0.284114
""	5 Discourse Contour Recognition	0.370303	0.363469	0.370303	0.000000
bert-base-uncased	1 Stance Alignment	0.498961	0.498374	0.465699	0.033262
""	1_5 Stance Alignment light	0.629714	0.603981	0.478904	0.150810
""	2 Sequential Coherence	0.618077	0.606799	0.480417	0.137660
""	3 Interactive Dynamics	0.356050	0.340300	0.256788	0.099261
""	4 Claim Depth Hierarchy	0.486140	0.488249	0.472631	0.013509
""	4_5 Claim Depth Hierarchy light	0.766738	0.752188	0.473091	0.293647
""	5 Discourse Contour Recognition	0.376629	0.369770	0.189576	0.187052
facebook/bart-base	1 Stance Alignment	0.507599	0.506536	0.488170	0.019429
""	1_5 Stance Alignment light	0.640547	0.608924	0.480744	0.159802
""	2 Sequential Coherence	0.622160	0.609758	0.489282	0.132878
""	3 Interactive Dynamics	0.362727	0.351022	0.256866	0.105861
""	4 Claim Depth Hierarchy	0.477798	0.478579	0.488580	-0.010782
""	4_5 Claim Depth Hierarchy light	0.778337	0.760018	0.485486	0.292851
""	5 Discourse Contour Recognition	0.388624	0.388751	0.388625	-0.000001
gpt2	1 Stance Alignment	0.509120	0.510731	0.480396	0.028724

""	1_5 Stance Alignment light	0.628942	0.606344	0.485612	0.143331
""	2 Sequential Coherence	0.622003	0.610002	0.485370	0.136633
""	3 Interactive Dynamics	0.352357	0.343110	0.256885	0.095472
""	4 Claim Depth Hierarchy	0.481703	0.484708	0.480488	0.001214
""	4_5 Claim Depth Hierarchy light	0.767588	0.727342	0.486089	0.281498
""	5 Discourse Contour Recognition	0.383902	0.374369	0.345420	0.038483
microsoft/deberta-v3-base	1 Stance Alignment	0.511299	0.503955	0.495724	0.015575
""	1_5 Stance Alignment light	0.665563	0.598844	0.476081	0.189482
""	2 Sequential Coherence	0.624821	0.603041	0.491028	0.133793
""	3 Interactive Dynamics	0.365508	0.352382	0.256948	0.108559
""	4 Claim Depth Hierarchy	0.475092	0.479983	0.488811	-0.013719
""	4_5 Claim Depth Hierarchy light	0.791698	0.780747	0.475414	0.316284
""	5 Discourse Contour Recognition	0.377465	0.346870	0.377465	0.000000

6.4.1 LLM Rankings

	index	permutation	no_context	selectivity	robustness	final_overall_rank
4	microsoft/deberta-v3-base	1.0	4.0	1.0	1.0	1.75
1	bert-base-uncased	3.0	3.0	2.0	2.0	2.50
0	albert-base-v2	2.0	1.0	5.0	4.0	3.00
3	gpt2	4.0	2.0	4.0	3.0	3.25
2	facebook/bart-base	5.0	5.0	3.0	5.0	4.50

Figure 63 - LLMs Ranking - All Categories

7 Outlook

The work was exciting and challenging at the same time. As a fan of literature, art, and new inventions, I found the combination of new technology and current topics fascinating: balancing statistics, linguistics, data analysis, and LLMs requires a lot of counterbalance.

I would often have liked to delve deeper into certain parts of the work, such as the evaluation and the associated difficulties in specific tasks (mainly Task 2 and Task 5; further simplification would have been reasonable). It sometimes seemed that as soon as I reached a certain level of efficiency, which takes a little longer for more complex topics, the clock would call for the next phase and rudely wake me up.

It would have been interesting to see how current LLMs (December 2023) are again more optimized for following discussions, performing and differing tasks such as LLAMA2 or a newer version of GPT. The exponential progress in technology has again produced a lot during this semester. It makes this work seem outdated or bygone in some cases. However, whether LLMs can follow discussions and to what extent they can do so is still (for a long time?) not answered and will at least not be fully answered until the singularity of man and machine.

The iterative process leaves you open to creativity and flexibility, which unsettles you but simultaneously moves you and keeps your curiosity alive. Nevertheless, this also gave me an overview of the complexity, a first pass through a labyrinth that I could go through repeatedly at a higher difficulty level. The End of this work feels more like a beginning, a foundation on which I could continue.

8 Appendix

8.1 Complesis Task Definition

Version 13.06.2023 / bcl

Modul:	Dept I BAA HS23

Titel:	Can Language Models Follow Discussions?
Ausgangslage und Problemstellung:	Die Diskussion ist ein elementarer Bestandteil des Meinungsaustausches. Dies kann sowohl synchron in einem direkten Gespräch von mehreren Personen oder asynchron auf Social Media, Foren, oder Diskussions-Plattformen wie kialo.com geschehen. Dabei haben Personen oft ein implizites Verständnis über die aktuelle Granularität und Richtung der Diskussion zu einem gewissen Zeitpunkt. Jedoch ist es unklar, inwiefern Language Model (wie BERT (Devlin et al., 2018), T5 (Raffel et al., 2020), oder UL2 (Tay et al., 2022)) diese Kompetenzen auch besitzen. Innerhalb dieses Projektes sollen darum die Fähigkeiten von solchen Modellen Diskussionen zu folgen erprobt werden. Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Bahri, D., Schuster, T., ... & Metzler, D. (2022). Unifying language learning paradigms. arXiv preprint arXiv:2205.05131. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1), 5485-5551.
Ziel der Arbeit und erwartete Resultate:	Das Ziel dieser Arbeit ist zum einen relevanten theoretischen Hintergründe wie Rethorical Structure Theory (Mann and Thompson, 1987) oder Discourse Representation Theory (Kamp et al., 2011; Geurts and Beaver, 2007) zu erarbeiten. Darauf aufbauend sollen verschiedene Tasks definiert werden welche die nötigen Kompetenzen repräsentieren, um einer Diskussion zu folgen (z.b. betreffen zwei Aussagen das gleiche Thema). Abschliessend sollen verschiedenen aktuelle Modelle mithilfe von Probing (Tenney et al., 2019; Rogers et al., 2021) oder Prompting (Liu et al., 2023) auf diesen Tasks evaluiert werden.

	Damit soll die folgende Hauptforschungsfrage beantwortet werden, wie gut können Language Models Diskussionen folgen? Dazu werden die folgenden Unter-Forschungsfragen betrachtet: - Welche Eigenschaften sind essenziell, um Diskussionen folgen zu können? - Wie können diese Eigenschaften mithilfe von Probing Task innerhalb von Language Models verifiziert werden? - Wie unterscheiden sich verschiedene Language Models auf diesen Tasks und welche ihrer Eigenschaften sind entscheidend für Unterschiede? - Welches implizites Verständnis einer Diskussion können wir von einem Language Model erwarten? Resultierend aus dieser Arbeit werden folgende Resultate erwartet: - Theoretische Erarbeitung von technischen (Natural Lanugage Processing, Machine Learning) und theoretischen Aspekten (welche Fähigkeiten sind nötig, um einer Diskussion zu folgen?) - Definition von 5+ Tasks um Language Models auf diesen Fähigkeiten zu überprüfen - Evaluation von verschiedenen Language Models anhand von diesem definierten Task - Bericht - Repository mit dem verwendeten Code Mann, W. C., & Thompson, S. A. (1987). Rhetorical structure theory: A theory of text organization (pp. 87-190). Los Angeles: University of Southern California, Information Sciences Institute. Geurts, B., & Beaver, D. (2007). Discourse representation theory. Kamp, H., Van Genabith, J., & Reyle, U. (2011). Discourse representation theory. Handbook of Philosophical Logic: Volume 15, 125-394. Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. arXiv preprint arXiv:1905.05950. Rogers, A., Kovaleva, O., & Rumshisky, A. (2021). A primer in BERTology: What we know about how BERT works. Transactions of the Association for Computational Linguistics, 8, 842-866. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys, 55(9), 1-35.
Gewünschte Methoden, Vorgehen:	Iterative Forschung, Model Analysis, Probing

Kreativität, Methoden, Innovation:	Die Aufgabenstellung bietet sehr viel Platz für Kreativität sowohl in der theoretischen Erarbeitung als auch in der methodischen und technischen Umsetzung. Publikation der Arbeit in der Form eines Artikels ist möglich - kein muss.
Sonstige Bemerkungen:	Die nötigen Daten werden zur Verfügung gestellt. Python, Hugginface-Transformers, Pytorch, Erfahrungen mit NLP.

Projektteam

Student:in 1:	Nico Previtali
Betreuer:in:	Waldis

Auftraggeber

Firma:	HSLU
Ansprechperson:	Andreas Waldis
Funktion:	Wissenschaftlicher Mitarbeiter / Doktorand
Strasse:	
PLZ/Ort:	
Telefon:	0798331719
E-Mail:	andreas.waldis@hslu.ch
Website:	

8.2 Notebooks and Data

- All notebooks and WANDB logging results are stored in a private Github repository at the following web address:
  https://github.com/digwit678/Can-Language-Models-Follow-Discussions
- As the files are too large for Github, the final folds.csv files with the probing data are still stored on Juypterlab where they were run.

9 Table of Figures

10 References

This thesis has been reviewed by GPT-4 and DeepL translator.

Adolphs, R. (2010). What does the amygdala contribute to social cognition? *Annals of the New York Academy of Sciences*, *1191*(1), 42–61. https://doi.org/10.1111/j.1749-6632.2010.05445.x

albert-base-v2 · Hugging Face. (n.d.). Huggingface.co. https://huggingface.co/albert-base-v2

Alessio Miaschi, Sarti, G., Brunato, D., Dell'Orletta, F., & Venturi, G. (2022). Probing Linguistic Knowledge in Italian Neural Language Models across Language Varieties. *IJCoL*, *8*(1). https://doi.org/10.4000/ijcol.965

Alhassan, A., Zhang, J., & Schlegel, V. (2022). *"Am I the Bad One"? Predicting the Moral Judgement of the Crowd Using Pre-trained Language Models* (pp. 20–25). https://aclanthology.org/2022.lrec-1.28.pdf

Bateman, J. A., & Delin, J. (2006). Rhetorical Structure Theory. *Elsevier EBooks*, 589–597. https://doi.org/10.1016/b0-08-044854-2/00541-1

BERT — transformers 3.0.2 documentation. (n.d.). Huggingface.co. https://huggingface.co/transformers/v3.0.2/model_doc/bert.html

Davies, B. L. (2007). Grice's Cooperative Principle: Meaning and rationality. *Journal of Pragmatics*, *39*(12), 2308–2331. https://doi.org/10.1016/j.pragma.2007.09.002

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018, October 11). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. ArXiv.org. https://arxiv.org/abs/1810.04805

ELECTRA — transformers 3.0.2 documentation. (n.d.). Huggingface.co. Retrieved January 3, 2024, from https://huggingface.co/transformers/v3.0.2/model_doc/electra.html

facebook/bart-base · Hugging Face. (n.d.). Huggingface.co. https://huggingface.co/facebook/bart-base

gpt2 · Hugging Face. (n.d.). Huggingface.co. https://huggingface.co/gpt2

Hewitt, J., & Liang, P. (2019, September 7). *Designing and Interpreting Probes with Control Tasks*. ArXiv.org. https://doi.org/10.48550/arXiv.1909.03368

Huber, P., & Carenini, G. (2022a). Towards Understanding Large-Scale Discourse Structures in Pre-Trained and Fine-Tuned Language Models. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. https://doi.org/10.18653/v1/2022.naacl-main.170

Huber, P., & Carenini, G. (2022b). Towards Understanding Large-Scale Discourse Structures in Pre-Trained and Fine-Tuned Language Models. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. https://doi.org/10.18653/v1/2022.naacl-main.170

Kamp, H., Van Genabith, J., & Reyle, U. (2010). Discourse Representation Theory. *Handbook of Philosophical Logic*, 125–394. https://doi.org/10.1007/978-94-007-0485-5_3

Kim, N., & Schuster, S. (2023). *Entity Tracking in Language Models*. https://doi.org/10.18653/v1/2023.acl-long.213

Koto, F., Lau, J. H., & Baldwin, T. (2021, April 12). *Discourse Probing of Pretrained Language Models*. ArXiv.org. https://doi.org/10.48550/arXiv.2104.05882

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods*

in Natural Language Processing. https://doi.org/10.48550/arxiv.2107.13586

microsoft/deberta-v3-base · Hugging Face. (n.d.). Huggingface.co. https://huggingface.co/microsoft/deberta-v3-base

Murathan Kurfalı, & Östling, R. (2021). Probing Multilingual Language Models for Discourse. *ArXiv (Cornell University)*. https://doi.org/10.18653/v1/2021.repl4nlp-1.2

OpenAI GPT2 — transformers 3.0.2 documentation. (n.d.). Huggingface.co. Retrieved January 3, 2024, from https://huggingface.co/transformers/v3.0.2/model_doc/gpt2.html#overview

Premack, D., & Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, *1*(4), 515.

Rajmohan, V., & Mohandas, E. (2007). Mirror neuron system. *Indian Journal of Psychiatry*, *49*(1), 66. https://doi.org/10.4103/0019-5545.31522

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, *8*, 842–866. https://doi.org/10.1162/tacl_a_00349

Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., Chung, H. W., Shakeri, S., Bahri, D., Schuster, T., Zheng, H. S., Zhou, D., Houlsby, N., & Metzler, D. (2023, February 28). *UL2: Unifying Language Learning Paradigms*. ArXiv.org. https://doi.org/10.48550/arXiv.2205.05131

Tenney, I., Das, D., & Pavlick, E. (2019). *BERT Rediscovers the Classical NLP Pipeline*. https://doi.org/10.48550/arxiv.1905.05950

Tran, K. H., McDonald, A. P., D'Arcy, R. C., & Song, X. (2021). Contextual Processing and the Impacts of Aging and Neurodegeneration: A Scoping Review. *Clinical Interventions in Aging*, *Volume 16*, 345–361. https://doi.org/10.2147/cia.s287619

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, June 12). *Attention Is All You Need*. ArXiv.org. https://arxiv.org/abs/1706.03762

Wang, Y., Li, S., & Yang, J. (2018, October 1). *Toward Fast and Accurate Neural Discourse Segmentation*. ACLWeb; Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1116

Wikipedia Contributors. (2019a, January 25). *Mann–Whitney U test*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test

Wikipedia Contributors. (2019b, November 15). *Cooperative principle*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Cooperative_principle

Xu, J., Gan, Z., Cheng, Y., & Liu, J. (2019). *Discourse-Aware Neural Extractive Text Summarization*. https://doi.org/10.48550/arxiv.1910.14142