Master Degree in Computer Science

Master Degree in Data Science for Economics and Health

Natural Language Processing

Prof. Alfio Ferrara

Dott. Sergio Picascia, Dott.ssa Elisabetta Rocchetti

Department of Computer Science, Università degli Studi di Milano
Room 7012 via Celoria 18, 20133 Milano, Italia alfio.ferrara@unimi.it

Ideas for final projects

Instructions

The final project consists in the preparation of a short study on one of the topics of the course, identifying a precise research question and measurable objectives. The project will propose a methodology for solving the research question and provide an experimental verification of the results obtained according to results evaluation metrics. The emphasis is not on obtaining high performance but rather on the critical discussion of the results obtained in order to understand the potential effectiveness of the proposed methodology.

The results must be documented in a short article of not less than 4 pages and no more than 8, composed according to the guidelines available here: template and using the corresponding $LaTeX$ or MS Word templates. Students have also to provide access to a GitHub repository containing the code and reproducible experimental results.

Finally, the project will be discussed after a **10 minutes presentation in English with slides**.

Procedure

Exam dates are just for the registration of the final grade. The project discussion will be set by appointment, according to the following procedure:

1. Subscribe to any available date
2. Contact Prof. Ferrara as soon as
    1. The project is finished and ready to be discussed
    2. After the date of your subscription is expired
3. Setup an appointment and discuss your work

When contacting Prof. Ferrara for the appointment, **provide the following information**:

1. The exam date you are subscribed to
2. The pdf version of your report
3. The link to the GitHub repository containing the code for the project

Example: you subscribe the exam date of [Month] [Day]. **Anytime after [Month] [Day]**, when the **project is ready**, you will contact Prof. Ferrara and set an appointment. You discuss the project during the appointment.

If you are **interested in doing your final master thesis on these topics**, the final project may be a preliminary work in view of the thesis. In this case, discuss the contents with Prof. Ferrara during the project discussion.

Structure of the paper

1. **Introduction**

   Provides an overview of the project and a short dicsussion on the pertinent literature

2. **Research question and methodology**

   Provides a clear statement on the goals of the project, an overview of the proposed approach, and a formal definition of the problem

3. **Experimental results**

   Provides an overview of the dataset used for experiments, the metrics used for evaluating performances, and the experimental methodology. Presents experimental results as plots and/or tables

4. **Concluding remarks**

   Provides a critical discussion on the experimental results and some ideas for future work

AI Usage Disclaimer

Parts of this projects have been developed with the assistance of **OpenAI’s ChatGPT (GPT-5)**. The AI was used to support the **development of project ideas, the structuring of methodological workflows, the drafting of descriptive texts**, and the **identification of relevant datasets and references**. All content produced with AI assistance has been **carefully reviewed, edited, and validated** by me. I take full responsibility for the final content and its accuracy, relevance, and academic integrity.

Using AI (for students)

Generative AI tools (such as ChatGPT, Claude, Mistral, or similar models) **may be used in this project**, both as an object of investigation and as a tool to support the development process. Students are encouraged to explore how these models function, interact with them creatively, and leverage them as **inspiration or assistance in ideation, drafting, or experimentation**.

However, **AI should not be used as a substitute for original work**. The responsibility for the structure, reasoning, and understanding of the project remains entirely with the student.

If generative AI has been used at any stage of the project, it is **mandatory to include a disclaimer** clearly specifying:

- **Which models** have been used (e.g., GPT-4, Claude 3, etc.)
- **For what purposes** (e.g., drafting text, summarizing ideas, generating code or examples)
- **To what extent** the outputs were modified, verified, or integrated into the final submission

The project will be assessed not only based on its output, but also on the **student's ability to explain and justify all choices made**. A final **interview will evaluate the depth of understanding**, and any lack of clarity or over-reliance on AI-generated material without proper insight may negatively affect the evaluation.

Generative AI should be seen as a **creativity support tool**, not as a replacement for critical thinking, problem solving, or technical development.

Instructions on coding

All project code should be written with clarity, modularity, and reusability in mind. The implementation should **not consist of a single large notebook**, but rather follow a structured and maintainable design. The recommended practice is to organize the logic into **Python modules and packages**, using **object-oriented programming (OOP)** principles where appropriate (e.g., defining classes for models, datasets, or evaluation pipelines).

Jupyter notebooks should be used primarily for **demonstration, experimentation, and visualization**, not for hosting the full application logic. Each notebook should import and showcase components from the main Python modules, illustrating how they work in practice. Code should include meaningful docstrings, comments, and clear function signatures.

Where possible, students are encouraged to separate concerns into layers — e.g., data loading and preprocessing, model interface, evaluation, and visualization — to facilitate testing and future reuse. The resulting repository should be **clean, reproducible, and extensible**, allowing others to replicate results or build upon the developed framework.

Project ideas

The following are ideas for projects. For each idea, a short description, example of datasets that can be used, and bibliographic references are provided. Students may **choose one of the following** as their project theme or **they can propose their own idea**, structuring the proposal as those presented in this document. In the latter case, just send the project description to Prof. Ferrara.

Project are organized in themantic clusters. The methodological notes, the datasets and the references are intended exclusively as a starting guideline. Students are encouraged to find their own data and/or references when needed and they can provide their personal interpretation to the methodological suggestions.

Thematic Cluster 1: Reasoning, Logic & Cognition

This cluster investigates the reasoning and cognitive capacities of Large Language Models (LLMs). Projects in this group focus on truthfulness, logical inference, and the interaction between symbolic and linguistic knowledge, analysing how models handle contradiction, abstraction, and consistency in multi-step reasoning.

P1. Truth, Lies, and Reasoning Machines

This project investigates how Large Language Models (LLMs) reason when exposed to **false, incomplete, or contradictory information**. The central goal is to understand whether these models can detect inconsistencies, resist misinformation, and maintain logical coherence during multi-step reasoning. The project aims at exploring both the *fragility* and *resilience* of reasoning in LLMs under “truth distortion” scenarios.

Core Pipeline Sketch:
The workflow involves constructing controlled reasoning datasets that include factual, counterfactual, and contradictory cases. Each model is prompted under these conditions, and the resulting reasoning chains are extracted, visualized, and compared using factuality and logical-consistency metrics.

Expected Outcomes:
Students will quantify how logical coherence deteriorates under misinformation stress and evaluate whether explicit self-verification prompts help preserve reasoning integrity across tasks.

Methodology

1. **Design reasoning tasks** involving factual and counterfactual statements (e.g., “If Paris were in Italy…”), or inject controlled falsehoods into multi-hop question answering datasets.
2. **Compare model behaviors** across setups: baseline (factual), noisy (contradictory), and adversarial (deliberately misleading).
3. Apply **explainability tools** (e.g., attention visualization, token attribution, or probability tracing) to analyze *where* and *how* the model deviates from truthful inference.
4. Implement **self-correction or verification prompts** (“Are you sure?”, “Check your assumptions”) to assess the model’s capacity for introspective reasoning.
5. **Evaluate outputs** using logical validity metrics, factual accuracy, and qualitative inspection of reasoning chains.

Students are encouraged to analyze the *types of reasoning failures* (e.g., belief persistence, hallucination propagation, circular logic) and discuss how truth distortion affects reasoning reliability.

Dataset

- TruthfulQA: Measuring How Models Mimic Human Falsehoods. https://github.com/sylinrl/TruthfulQA
- HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering.https://hotpotqa.github.io/

References

- Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). Locating and editing factual associations in gpt. Advances in neural information processing systems, 35, 17359-17372.
- Lin, S., Hilton, J., & Evans, O. (2022, May). TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 3214-3252).
- Zhou, X., Wang, Q., Wang, X., Tang, H., & Liu, X. (2023). Large language model soft ideologization via AI-self-consciousness. *arXiv preprint arXiv:2309.16167*.

P2. The Knowledge Translator

This project investigates whether **Large Language Models (LLMs)** can act as *semantic interpreters* for structured knowledge queries. Given a **formal query** (e.g., expressed in logical form, SPARQL, or RDF triples) and a **textual corpus** as the only knowledge source, the model is asked to retrieve, synthesize, and justify answers using natural language understanding rather than database reasoning. The goal is to explore how far LLMs can go in “translating logic into language”, connecting symbolic intent with unstructured knowledge.

Core Pipeline Sketch:
Implementation begins with structured queries derived from an ontology, which are paired with unstructured textual corpora. The LLM interprets each query and returns variable bindings based on textual evidence, later compared against gold-standard knowledge base results.

Expected Outcomes:
The project should provide empirical insight into the capacity of LLMs to bridge symbolic queries and natural language reasoning, revealing how linguistic inference compensates for missing structured representations.

Methodology

1. **Formal Query Definition:** Select or design a set of structured queries derived from an existing ontology or knowledge graph (e.g., `?author wrote ?book WHERE book.genre = ‘science fiction’`).
2. **Textual Knowledge Source:** Prepare a corpus (e.g., Wikipedia articles, domain-specific texts) containing relevant information but not in structured form.
3. **Model Task Design:** Experiment with prompting or fine-tuning strategies that guide the LLM to “simulate” logical reasoning: a) Identify entities and relations in text that correspond to the predicates in the query. b) Extract possible variable assignments and output them in a structured format (e.g., JSON or table).
4. **Evaluation and Comparison:**
    - Measure **semantic correctness** against baselines such as symbolic query execution or retrieval-based systems.
    - Analyze **error types**: misunderstanding of logical operators, entity mismatch, or overgeneralization

Dataset

- Wikidata / DBpedia – to generate reference triples and queries for evaluation.
- Wikipedia corpus – as the textual grounding source.
- MetaQA – benchmark for multi-hop question answering over knowledge graphs (can be adapted for text-based inference).
- Optional: domain-specific corpora (e.g., scientific articles, biographies, or cultural heritage text collections).

References

- Saeed, Mohammed, Nicola De Cao, and Paolo Papotti. "Querying large language models with SQL." *arXiv preprint arXiv:2304.00472* (2023).
- Badaro, G., Saeed, M., & Papotti, P. (2023). Transformers for tabular data representation: A survey of models and applications. *Transactions of the Association for Computational Linguistics*, *11*, 227-249.
- Ngonga Ngomo, A. C., Bühmann, L., Unger, C., Lehmann, J., & Gerber, D. (2013, May). Sorry, i don't speak SPARQL: translating SPARQL queries into natural language. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 977-988).

Thematic Cluster 2: Agentic Behavior & Interaction

This cluster explores the emergent agency of LLMs in interactive or multi-agent settings. It studies negotiation, cooperation, competition, and coordination between artificial agents, providing insights into pragmatic communication, strategic behaviour, and collective intelligence in language-based systems.

P3. The Negotiation Arena

This project investigates how **Large Language Models (LLMs)** behave as autonomous agents engaged in negotiation, cooperation, or strategic dialogue. Two or more models are placed in simulated scenarios where they must **reach an agreement, trade resources, or align on decisions** despite having distinct goals or incomplete information. The objective is to analyze the **emergent communicative strategies**—such as persuasion, concession, deception, or cooperation—and to evaluate whether these behaviors reflect genuine reasoning, pragmatic adaptation, or scripted imitation.

Core Pipeline Sketch:
Agents are instantiated with distinct goals or utility functions and engage in multi-round conversations to reach agreements. Simulations log all exchanges, which are evaluated quantitatively (agreement rate, utility gain) and qualitatively (linguistic persuasion strategies).

Expected Outcomes:
Students will uncover how cooperative or adversarial behaviours emerge among LLMs, identifying pragmatic and linguistic features correlated with success or failure in negotiation.

Methodology

1. **Scenario Design**: Define one or more negotiation settings such as resource division (“splitting items or money”), task scheduling (“allocating responsibilities”), or preference alignment (“choosing the best option for both parties”). Each agent receives private information or asymmetric incentives encoded in its prompt.
2. **Agent Configuration**: Instantiate two or more LLM agents with distinct “personas” or objectives. Examples: *Agent A seeks maximum profit*, *Agent B values fairness*, *Agent C minimizes risk*. Optionally, include an adjudicator model (or a human evaluator) to judge outcomes.
3. **Dialogue Simulation**: Implement iterative rounds of conversation where agents exchange proposals until an agreement or impasse is reached. We may have test variations such as **Cooperative mode** (shared goal), **Competitive mode** (conflicting goals) or **Mixed mode** (partial cooperation or deception allowed).

4. **Analysis and Metrics**: Measure **agreement rate**, **rounds to convergence**, **utility scores**, and **language complexity**. Qualitatively analyze dialogue transcripts for persuasion tactics, emotional tone, and logical coherence.

Dataset

No fixed dataset required; negotiation scenarios can be **synthetically generated** or adapted from existing dialogue datasets.

References

- Lewis, M., Yarats, D., Dauphin, Y., Parikh, D., & Batra, D. (2017, September). Deal or No Deal? End-to-End Learning of Negotiation Dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 2443-2453).
- Akin, S., Tiwari, S. T., Bhattacharya, R., Raman, S. A., Mohanty, K., & Krishnan, S. (2025). Socialized Learning and Emergent Behaviors in Multi-Agent Systems based on Multimodal Large Language Models. *arXiv preprint arXiv:2510.18515*.
- Gupta, P., Zhong, Q., Yakura, H., Eisenmann, T., & Rahwan, I. (2025). The Role of Social Learning and Collective Norm Formation in Fostering Cooperation in LLM Multi-Agent Systems. *arXiv preprint arXiv:2510.14401*.

P4. Game of Thoughts

This project explores how **Large Language Models (LLMs)** understand, manipulate, and generate **structured rule systems** — from interpreting existing board games to inventing entirely new ones.
By treating games as a proxy for structured reasoning, the goal is to evaluate the model's capacity for **logical consistency, creativity, and procedural understanding**. Students will experiment with both *rule comprehension* and *rule creation* tasks, observing how LLMs balance freedom and constraint in structured reasoning contexts.

Core Pipeline Sketch:
The pipeline covers rule comprehension, simulation, and creative generation. Models interpret existing game rules, simulate valid moves from intermediate states, and generate new playable games. Evaluation focuses on internal consistency and procedural validity.

Expected Outcomes:
Students will document how LLMs balance creativity and logical constraint, offering a comparative view of the models' ability to internalise and extend structured rule systems.

Methodology

1. **Game Understanding**: Provide the model with natural-language rulebooks of real games (see (BGG)[https://boardgamegeek.com/]). Ask the model to: explain the rules in simplified form, identify missing, ambiguous, or inconsistent rules, suggest corrections or clarifications.
2. **Game Simulation**: Feed the model a **current game state** (expressed in natural language) and ask for the next valid move or a strategic suggestion. Evaluate whether the model respects the rule set and whether its decisions remain coherent across turns.

3. **Game Generation**: In a creative extension, instruct the model to **design a new game** given a theme or constraint (e.g., “a cooperative card game about climate change”). Assess the originality, coherence, and playability of the generated rule set.
4. **Evaluation**: Compare the model’s understanding with human-readable ground truth (e.g., actual rules or expected moves). Evaluate generated games via heuristic criteria: *internal consistency*, *balance*, *clarity*, and *fun factor*. Optionally, implement a parser or lightweight simulator to test the model’s generated rules in action.

Dataset

- BoardGameGeek (BGG) API — metadata and rule summaries for thousands of board games.
- Official rulebooks freely available online from publishers.
- Ludii Game Database — structured repository of formalized game rules.

References

- Todd, G., Padula, A. G., Stephenson, M., Piette, É., Soemers, D. J., & Togelius, J. (2024). Gavel: Generating games via evolution and language models. *Advances in Neural Information Processing Systems*, *37*, 110723-110745.
- Hu, C., Zhao, Y., & Liu, J. (2024, August). Game generation via large language models. In *2024 IEEE Conference on Games (CoG)* (pp. 1-4). IEEE.
- Piette, E., Stephenson, M., Soemers, D. J., & Browne, C. (2021, August). General board game concepts. In *2021 IEEE Conference on Games (CoG)* (pp. 01-08). IEEE.

Thematic Cluster 3: Knowledge, Retrieval & Robustness

This cluster focuses on the reliability and structure of knowledge in LLMs and retrieval-augmented systems. Projects analyse the robustness of retrieved information, the process of knowledge compression or distillation, and the mechanisms by which models verify, maintain, or adapt factual content across domains.

P5. In RAG We Trust?

This project evaluates how **Retrieval-Augmented Generation (RAG)** systems manage source credibility and factual reliability when retrieving information from multiple documents. Instead of designing new RAG architectures, students will focus on **quantifying and analyzing reliability** by measuring how models weigh and reconcile conflicting or falsified sources.

Core Pipeline Sketch:
A RAG pipeline is constructed with an intentional “poisoned” retrieval component. The system’s robustness is tested against false or conflicting evidence while models are prompted to assess source reliability. Results are analysed via factual accuracy and self-consistency measures.

Expected Outcomes:
The project will yield a quantitative assessment of how retrieval-augmented generation systems handle misinformation, producing metrics and qualitative insights on trust and verification mechanisms.

Methodology

1. **Controlled Data Poisoning**: Introduce intentional falsehoods or contradictions into retrieved document sets.
2. **Multi-source Verification**: Measure whether the model can identify inconsistencies, seek confirmation across multiple documents, or hedge its answers.
3. **Prompt Design**: Test whether meta-prompts (e.g., “check consistency across documents”) improve reliability.
4. **Evaluation**: Compare factual accuracy, hallucination rate, and self-consistency across models and retrieval settings.

Dataset

- FEVER Dataset for fact-checking and verification.
- HotpotQA multi-hop QA benchmark for evidence reasoning.
- Synthetic datasets created by deliberately injecting false or contradictory passages.

References

- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, *33*, 9459-9474.
- Zhou, Y., Liu, Y., Li, X., Jin, J., Qian, H., Liu, Z., ... & Yu, P. S. (2024). Trustworthiness in retrieval-augmented generation systems: A survey. *arXiv preprint arXiv:2409.10102*.
- Singal, R., Patwa, P., Patwa, P., Chadha, A., & Das, A. (2024, November). Evidence-backed fact checking using RAG and few-shot in-context learning with LLMs. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)* (pp. 91-98).

P6. The Apprentice Model

This project explores **knowledge distillation** by training a smaller model to imitate a larger LLM on a specific domain or reasoning task. The aim is to obtain lightweight, domain-specialized “expert” models while analyzing trade-offs between **efficiency, specialization, and generalization**.

Core Pipeline Sketch:
A smaller model is trained using outputs or intermediate representations from a teacher LLM. The student model’s predictions are compared with the teacher’s across various benchmarks to evaluate compression and transfer efficiency.

Expected Outcomes:
Students will measure trade-offs between model compactness, domain specialisation, and reasoning quality, demonstrating the degree to which distilled systems retain expert knowledge.

Methodology

1. **Data Collection**: Use a large LLM to generate or label examples in a specific domain (e.g., medical,

legal, or cultural) and use the large LLM as a teacher model.

2. **Model Distillation**: Train a smaller transformer (e.g., DistilBERT, TinyT5 or a custom model) to reproduce the teacher model's predictions or reasoning chains.
3. **Evaluation**: Compare distilled vs. teacher model on domain benchmarks, focusing on performance drop, interpretability, and computational savings.
4. **Extension**: Experiment with multi-teacher distillation or domain adaptation through selective fine-tuning.

Dataset

- Domain-specific datasets.
- Custom datasets generated by prompting larger models (e.g., GPT-4, Claude).

References

- Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Ji, G., & Zhu, Z. (2020). Knowledge distillation in wide neural networks: Risk bound, data efficiency and imperfect teacher. *Advances in Neural Information Processing Systems*, *33*, 20823-20833.
- Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., ... & Liu, Q. (2020, November). Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020* (pp. 4163-4174).

P7. Analyzing Thematic Alignment in Scientific Journals

The core objective of the project is to quantitatively assess whether the articles published in a specific journal align with its stated "Aims & Scope". Students will develop a methodology to model the journal's intended focus and compare it against the content of its publications, potentially identifying thematic drift over time or outlier papers.

Core Pipeline Sketch:
Articles from a target journal are collected and represented as embeddings or topic distributions. The “Aims & Scope” section serves as a thematic reference, and alignment scores are computed for each paper to detect drift or anomalies.

Expected Outcomes:
Students will quantify thematic coherence within academic publishing, identifying outlier papers and trends that reveal long-term conceptual evolution or misalignment in editorial focus.

Methodology

- **Data Curation:** Select a scientific journal with a clearly defined "Aims & Scope" statement (e.g., from the journal's website). The "Aims & Scope" text will serve as the ground truth for the journal's intended focus.
- **Modelling the Content:** Process the raw text to distill its core subjects and meaning. Students must create a structured, machine-readable representation of the content for both the benchmark "Aims &

Scope" statement and for each article abstract in the corpus. This involves transforming unstructured text into a format that captures its primary themes and concepts, allowing for computational comparison.

- **Measure Alignment:** Design and implement a computational method to systematically measure the thematic overlap between each article and the journal's stated scope. This will involve using the representations created in the previous step to generate a quantitative "alignment score" for every paper in the corpus, indicating how closely its content matches the journal's mission.
- **Report Findings:** Analyze the distribution of alignment scores across the corpus and visualize the results, detecting potential "thematic drift" or identifying significant outliers. The analysis must be complemented by a qualitative inspection of the highest- and lowest-scoring articles to validate the metric and provide a nuanced interpretation of the findings.

Dataset

- Use a relevant API (like arXiv, Semantic Scholar, or PubMed), collect a corpus of article abstracts published in that journal over a period of time (e.g., the last 5-10 years).

References

- Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv preprint arXiv:2203.05794.
- Reimers, N., & Gurevych, I. (2019, November). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3982-3992).
- Picascia, S., Montanelli, S., Salini, S., & Verzillo, S. (2025). The Atlas of Data Science Research. IEEE Access.
- Hassan-Montero, Y., Guerrero-Bote, V. P., & De-Moya-Anegón, F. (2014). Graphical interface of the Scimago Journal and Country Rank: an interactive approach to accessing bibliometric information. El profesional de la información, 23(3).

Thematic Cluster 4: Prompt Engineering, Meta-NLP & Instruction Tuning

This cluster examines how the formulation of prompts and fine-tuning procedures shape model behaviour. It investigates adaptive prompting, meta-learning, and causal inference to better understand how models internalise instructions, optimise responses, and modify their linguistic and cognitive patterns.

P8. Evolution of a Prompt

This project designs an **automated prompt optimization framework**, where prompts evolve iteratively based on performance feedback. The goal is to study **prompt sensitivity** and develop adaptive, self-improving strategies that balance human control and model autonomy.

Core Pipeline Sketch:
A feedback-driven optimisation loop iteratively refines prompts using performance metrics or LLM self-evaluation. Each new prompt version is tested on target tasks, and improvements are monitored over generations.

Expected Outcomes:
The project will demonstrate how prompt evolution influences task performance and linguistic structure, producing comparative evidence of emergent meta-learning behaviours in LLMs.

Methodology

1. **Task Selection**: Choose a concrete NLP task (e.g., summarization, reasoning, or classification).
2. **Prompt Optimization Loop**: Implement iterative prompt mutation via scoring (e.g., accuracy, coherence, or BLEU).
3. **Feedback Mechanism**: Use either external metrics or LLM self-evaluation to guide evolution.
4. **Comparison**: Benchmark adaptive prompting against manually engineered baselines.

Dataset

- BIG-bench for challenging reasoning tasks.
- natural-instructions of language instructions for LLMs.

References

- Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., ... & Resnik, P. (2024). The prompt report: a systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*.
- Ye, Q., Ahmed, M., Pryzant, R., & Khani, F. (2024, August). Prompt engineering a prompt engineer. In *Findings of the Association for Computational Linguistics: ACL 2024* (pp. 355-385).
- Hsieh, C. J., Si, S., Yu, F., & Dhillon, I. (2024, August). Automatic engineering of long prompts. In *Findings of the Association for Computational Linguistics: ACL 2024* (pp. 10672-10685).

P9. Measuring total causal effects of instruction tuning on LLM behaviour

The process of "**instruction tuning**"—fine-tuning a base Large Language Model (LLM) on a dataset of instructions and desired outputs—is a critical step in creating helpful and safe AI assistants. While this process demonstrably improves a model's ability to follow commands, its broader impact on the model's underlying behavior is less understood. This project frames instruction tuning as a "treatment" and aims to measure its **total causal effect** on a range of model variables. The objective is to move beyond simple performance metrics and quantify the changes—both intended and unintended—that instruction tuning imparts on a model. The student will adopt a causal inference framework to compare an instruction-tuned model (the treatment group) with its pre-trained base version (the control group) to isolate and measure the effect of the tuning process itself.

Core Pipeline Sketch:
Two versions of the same model (base and instruction-tuned) are evaluated on predefined linguistic and behavioural variables. Causal inference techniques are applied to estimate total treatment effects of tuning.

Expected Outcomes:
Students will produce quantitative measures of how instruction tuning alters reasoning depth, bias expression, and linguistic features, establishing causal evidence for alignment processes.

Methodology

1. **Causal framework definition**: the *outcome* is a quantifiable output variable (see the next section for ideas on potential outcomes); the *treatment* is the application of instruction and alignment tuning to a base model; the *treatment group* is the set of output variable observations produced by the instruction-tuned model; the *control group* is the set of output variable observations produced by the base version of the model; the *total effect* is the measured difference in the outcome variable between the treatment and control groups, averaged over a diverse set of prompts
2. **Model and variable selection**: The student will select an open-source model family that provides public access to both a base and an instruction-tuned version. The student will choose one specific, measurable variable from the list in the next section to serve as the primary outcome for the experiment
3. **Dataset and prompt design**: A standardised set of prompts will be used to elicit behaviors related to the chosen variable. This can be sourced from existing academic benchmarks (e.g., for reasoning, toxicity, or bias) or be custom-designed to ensure a controlled experimental environment
4. **Experiments and outcome measurement**: The same set of prompts will be run through both the base and instruction-tuned models. A well-defined metric will be used to quantify the outcome (e.g., the simple logit or probability value, accuracy score, toxicity rating, refusal rate, linguistic complexity score).
5. **Analysis**: The total effect will be calculated by comparing the average metric scores between the two model versions. Statistical significance tests (e.g., t-tests) will be employed to determine if the observed effect is statistically meaningful. The outcome of this project will be a quantitative measure of how instruction-tuning causally affects a specific dimension of model behaviour

Potential variables for study (outcome variables)

The student may choose to investigate the effect of instruction tuning on variables such as:

- **Sycophancy:** The model's tendency to agree with a user's premise, even if it is factually incorrect.
- **Lexical Complexity:** The sophistication of the vocabulary used (e.g., measured by Flesch-Kincaid grade level).
- **Logical Reasoning:** Performance on standardized logical puzzles or benchmarks (e.g., sections from the LSAT, INVALSI).
- **Creative Output:** Divergence and novelty in creative writing or brainstorming tasks.
- **Political Bias:** The model's expressed political leaning across a spectrum of issues.
- **Gender or Professional Bias:** The prevalence of stereotypes in descriptions of people or professions.

Dataset

Any existing academic benchmarks (e.g., for reasoning, toxicity, or bias) or be custom-designed to ensure a controlled experimental environment (see also other experiments datasets).

References

- Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2022. Causal Inference in Natural Language Processing: Estimation, Prediction, Interpretation and Beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158.
- Vig, J.; Gehrmann, S.; Belinkov, Y.; Qian, S.; Nevo, D.; Singer, Y.; Shieber, S. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2020; Vol. 33, pp 12388–12401.
- Mats Faulborn, Indira Sen, Max Pellert, Andreas Spitz, and David Garcia. 2025. Only a Little to the Left: A Theory-grounded Measure of Political Bias in Large Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31684–31704, Vienna, Austria. Association for Computational Linguistics.

Thematic Cluster 5: Bias, Ethics & Cultural Intelligence

This cluster addresses the ethical, social, and cultural dimensions of LLM behaviour. Projects investigate moral alignment, value consistency, and cultural representation, studying how models interpret and express ethical stances or reproduce socio-linguistic biases.

P10. Right, Wrong, and Everything in Between

This project examines how LLMs respond to **morally ambiguous or ethically charged prompts**, analyzing *when, how,* and *why* models refuse, reframe, or justify their responses. The aim is to study the intersection between **moral alignment** and **linguistic pragmatics**, uncovering how ethical constraints shape the communicative behavior of models.

Core Pipeline Sketch:
Ethically ambiguous prompts are curated and submitted to multiple LLMs. Responses are analysed linguistically and semantically for refusal patterns, moral consistency, and affective stance.

Expected Outcomes:
The study will reveal how ethical alignment manifests in language, showing how models navigate moral grey areas and balancing safety constraints with expressiveness.

Methodology

1. **Scenario Construction**: Design ethically sensitive dialogues (e.g., fairness dilemmas, medical triage, privacy conflicts).
2. **Response Analysis**: Collect outputs from different LLMs and classify them by strategy (refusal, justification, reframing).

3. **Quantitative Evaluation**: Measure response diversity, moral consistency, and sentiment balance.
4. **Qualitative Analysis**: Compare linguistic markers (modality, politeness, uncertainty) across models and contexts.

Dataset

- ETHICS dataset: benchmarks for moral reasoning.
- Custom prompts reflecting social or political dilemmas.

References

- Hendrycks, D., Burns, C., Basart, S., Critch, A., Li, J., Song, D., & Steinhardt, J. (2020). Aligning ai with shared human values. *arXiv preprint arXiv:2008.02275*.
- Forbes, M., Hwang, J. D., Shwartz, V., Sap, M., & Choi, Y. (2020). Social chemistry 101: Learning to reason about social and moral norms. *arXiv preprint arXiv:2011.00620*.
- Bonagiri, V. K., Vennam, S., Govil, P., Kumaraguru, P., & Gaur, M. (2024). Sage: Evaluating moral consistency in large language models. *arXiv preprint arXiv:2402.13709*.

P11. Measuring total causal effects of prompting strategies on LLM behaviour

The way a query is formulated can significantly influence the output of a Large Language Model (LLM), especially on topics involving **cultural or contextual nuances**. This project applies a **causal framework** to measure the total effect of different prompting strategies on a model's linguistic and cultural expression. The core idea is to treat the prompt's formulation as a "treatment" and measure its impact on the model's response. The objective is to isolate and quantify how **two distinct prompting strategies lead to different model behaviors**. The experiment will compare responses from: (i) a *standard prompt* with no references to roles or languages; (ii) prompts that explicitly assign the model a cultural *persona* (e.g., "As a person from Nigeria..."); (iii) prompts written directly in a corresponding native *language* (e.g., Yoruba or Igbo). This allows for a precise measurement of the effect of *explicit role-playing* versus *implicit linguistic context*.

Core Pipeline Sketch:
Models are exposed to varying prompt formulations (neutral, persona-based, multilingual). The effect of each “treatment” on output bias and linguistic variation is measured using causal inference metrics.

Expected Outcomes:
Students will isolate the linguistic and cultural effects of prompt design, providing causal evidence of how context and framing shift model outputs and inferred values.

Methodology

1. **Causal framework definition**: the *outcome* is a quantifiable output variable (see the next section for ideas on potential outcomes); the *treatment* is the application of the selected prompting strategy either language or persona); the *treatment group* is the set of output variable observations produced by the selected prompting strategy; the *control group* is the set of output variable observations produced by the standard prompt; the *total effect* is the measured difference in the outcome variable

between the treatment and control groups, averaged over a diverse set of prompts

2. **Model and variable selection**: The student will select an open-source multilingual model. The student will choose one specific, measurable variable from the list in the next section to serve as the primary outcome for the experiment
3. **Dataset and prompt design**: A standardised set of prompts will be used to elicit behaviors related to the chosen variable. This can be sourced from existing academic benchmarks (e.g., for reasoning, toxicity, or bias) or be custom-designed to ensure a controlled experimental environment
4. **Experiments and outcome measurement**: Both sets of prompts will be run through the selected LLM, and the responses collected. A well-defined metric will be used to score the outputs based on the chosen variable (e.g., stereotype score, cultural knowledge accuracy, linguistic formality)
5. **Analysis**: The total effect of the prompting strategy will be calculated by comparing the average metric scores between the three groups. Statistical significance tests (e.g., t-tests) will be employed to determine if the observed effect is statistically meaningful

Potential variables for study (outcome variables)

The student may choose to investigate the effect of prompting strategy on variables such as:

- **Stereotype Prevalence:** The frequency of stereotypical associations versus nuanced descriptions.
- **Cultural Knowledge Accuracy:** The correctness of answers to questions about specific cultural facts or norms (see this benchmark)
- **Logical Reasoning:** Performance on standardized logical puzzles or benchmarks (e.g., sections from the LSAT, INVALSI).
- **Expression of Cultural Values:** How responses reflect cultural dimensions like individualism vs. collectivism.
- **Political Bias:** The model's expressed political leaning across a spectrum of issues.
- **Gender or Professional Bias:** The prevalence of stereotypes in descriptions of people or professions.

Dataset

- BLEnD: A Benchmark for LLMs on Everyday Knowledge in Diverse Cultures and Languages (NeurIPS 2024 Datasets and Benchmarks Track). Myung, J.; Lee, N.; Zhou, Y.; Jin, J.; Putri, R. A.; Antypas, D.; Borkakoty, H.; Kim, E.; Perez-Almendros, C.; Ayele, A. A.; Gutiérrez-Basulto, V.; Ibáñez-García, Y.; Lee, H.; Muhammad, S. H.; Park, K.; Rzayev, A. S.; White, N.; Yimam, S. M.; Pilehvar, M. T.; Ousidhoum, N.; Camacho-Collados, J.; Oh, A. BLEnD: A Benchmark for LLMs on Everyday Knowledge in Diverse Cultures and Languages. *Advances in Neural Information Processing Systems* **2024**, *37*, 78104–78146.
- ETHICS dataset: benchmarks for moral reasoning.
- Custom prompts reflecting social or political dilemmas.

References

- Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2022. Causal Inference in Natural Language Processing: Estimation, Prediction, Interpretation and Beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158.

- Vig, J.; Gehrmann, S.; Belinkov, Y.; Qian, S.; Nevo, D.; Singer, Y.; Shieber, S. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2020; Vol. 33, pp 12388–12401.
- Mats Faulborn, Indira Sen, Max Pellert, Andreas Spitz, and David Garcia. 2025. Only a Little to the Left: A Theory-grounded Measure of Political Bias in Large Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31684–31704, Vienna, Austria. Association for Computational Linguistics.

Thematic Cluster 6: Creativity, Narrative & Style

This cluster explores the expressive and generative capabilities of LLMs across creative, narrative, and stylistic domains. It examines how models emulate storytelling, produce cultural artefacts, and develop distinctive stylistic identities that blur the boundaries between computation and creativity.

P12. Stories We Tell (and the Machines Retell)

This project explores how **Large Language Models (LLMs)** represent and reproduce **narrative archetypes across different cultures and traditions**. By combining computational narratology, linguistic analysis, and network science, the project investigates whether storytelling structures — such as the *hero's journey*, *conflict–resolution arcs*, or *moral transformations* — appear as **universal cognitive patterns** or are culturally bound. Students will compare the **narrative logic of human-written texts** with **LLM-generated stories**, aiming to reveal both **shared deep structures** and **cultural biases** in automated narrative generation. Beyond textual comparison, the project also offers a creative perspective: it asks whether LLMs can act as "cultural translators" capable of adapting myths and stories across different symbolic systems.

Core Pipeline Sketch:
Narratives from multiple cultural traditions are modelled as structured event graphs. LLMs are prompted to retell these stories in cross-cultural contexts, and outputs are compared through graph similarity and semantic metrics.

Expected Outcomes:
The project will illustrate how storytelling patterns evolve across cultures and models, offering both computational and interpretive insight into narrative universals and biases.

Methodology

1. **Corpus Design and Data Collection**: Build or select a multilingual corpus of myths, folktales, or short stories from at least three cultural areas (e.g., European, East Asian, African, Indigenous). Optionally include both traditional texts and modern adaptations to observe continuity or drift in narrative motifs.

   Narrative Structure Modeling: Use NLP tools to extract narrative entities (characters, settings, key events) and relations (e.g., protagonist–antagonist, cause–effect, transformation arcs). Represent these elements as **narrative graphs** or event networks. Apply clustering or graph similarity analysis to detect recurring structures and motifs.

Cross-Cultural and Model Comparison: Ask LLMs (e.g., GPT-4, LLaMA, Claude) to generate retellings or analogues of specific myths in different cultural contexts. Evaluate whether the generated versions preserve, distort, or hybridize structural patterns (e.g., merging archetypes from different traditions).

Quantitative Analysis: Use metrics such as graph edit distance, motif overlap, or semantic role alignment to quantify structural similarity across corpora and models. Identify which narrative functions (e.g., quest, sacrifice, moral reward) are most stable or most distorted in generation.

Extensions (optional): Introduce multimodal inputs (e.g., images or symbolic motifs) to test whether models associate narrative meaning across modalities. Visualize narrative universes as cross-cultural story maps showing archetypal links and divergences.

Dataset

- LitBank. LitBank is an annotated dataset of 100 works of English-language fiction to support tasks in natural language processing and the computational humanities.
- World folktale corpora eventually scraped from the web or collected in other ways.

References

- Valls-Vargas, J., Zhu, J., & Ontañón, S. (2016). Predicting proppian narrative functions from stories in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (Vol. 12, No. 1, pp. 107-113).
- Kumaran, V., Rowe, J., & Lester, J. (2024, November). NARRATIVEGENIE: generating narrative beats and dynamic storytelling with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (Vol. 20, No. 1, pp. 76-86).
- Ranade, P., Dey, S., Joshi, A., & Finin, T. (2022). Computational understanding of narratives: A survey. *IEEE Access*, *10*, 101575-101594.

P13. The Aesthetics of Generation

This project investigates whether **Large Language Models (LLMs)** exhibit a distinctive **aesthetic or stylistic identity** in their generated outputs — a recognizable “voice” that persists across tasks, topics, and prompts. Beyond surface-level metrics of quality or fluency, the project asks: *do models have style?* Students will explore how lexical choices, syntactic patterns, rhythm, and discourse structure interact to form consistent stylistic fingerprints, and whether these can be identified, quantified, or even transferred across models. The study also encourages reflection on the philosophical dimension of style: what does “creativity” mean when it emerges from probabilistic generation rather than intention?

Core Pipeline Sketch:
Texts generated by different LLMs are collected under uniform prompts and analysed for stylistic and lexical variation. Stylometric features are extracted and visualised to identify distinct “voices” per model.

Expected Outcomes:
Students will characterise the stylistic identity of major LLMs, contributing to the emerging field of AI stylistics and authorship attribution.

Methodology

1. **Corpus Generation**: Generate a balanced corpus of model outputs (e.g., GPT-4, Claude, LLaMA, Mistral) across several genres: narration, argumentation, dialogue, and description. Keep prompts constant across models to ensure stylistic comparability.

   Stylometric Analysis: Extract quantitative stylistic features: lexical diversity, sentence length distribution, part-of-speech ratios, syntactic depth, punctuation frequency, and discourse markers. Apply clustering or dimensionality reduction (PCA, t-SNE) to visualize stylistic proximity between models.

   Comparative Evaluation: Use **authorship attribution** or **model identification** tasks to test whether a classifier can recognize which model produced a given text. Evaluate how robust these stylistic signatures remain when the same model is prompted in different tones or instructed to mimic a human author.

   Style Transfer and Transformation (Optional): Experiment with *style blending* or *cross-model paraphrasing*: can one model successfully emulate another's stylistic fingerprint? Analyze what linguistic transformations occur when style is transferred while maintaining semantic content.

   Visualization (Optional): Build visual maps of stylistic embeddings, showing clusters of models or genres as "aesthetic landscapes" in vector space.

Dataset

- Texts generated from open and closed LLMs (GPT, Claude, LLaMA, Mistral).
- Okulska, I., Stetsenko, D., Kołos, A., Karlińska, A., Głąbińska, K., & Nowakowski, A. (2023). Stylometrix: An open-source multilingual tool for representing stylometric vectors. *arXiv preprint arXiv:2309.12810*.

References

- Opara, C. (2024, July). StyloAI: Distinguishing AI-generated content with stylometric analysis. In *International conference on artificial intelligence in education* (pp. 105-114). Cham: Springer Nature Switzerland.
- Kumarage, T., Garland, J., Bhattacharjee, A., Trapeznikov, K., Ruston, S., & Liu, H. (2023). Stylometric detection of ai-generated text in twitter timelines. arXiv 2023. *arXiv preprint arXiv:2303.03697*.
- Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending against neural fake news. *Advances in neural information processing systems*, *32*.

Thematic Cluster 7: Explainability, Visualization & Model Understanding

This cluster investigates interpretability and transparency in language models. Projects aim to visualise and explain the internal dynamics of model reasoning, uncovering how abstract representations of meaning and attention evolve within high-dimensional linguistic spaces.

P14. Transparent Minds

This project aims to design and implement an **interactive toolkit for visualizing and interpreting the reasoning processes** of transformer-based language models. While most explainability studies remain abstract or static, this project focuses on creating a **hands-on, dynamic environment** where users can *see* and *experiment with* the inner mechanics of LLMs — attention flow, token importance, and activation patterns. The overarching goal is to make **model interpretability tangible** through visualization, enabling both qualitative exploration and quantitative assessment of interpretability methods.

Core Pipeline Sketch:
An interactive toolkit is developed for extracting and visualising internal states (attention weights, gradients, activations). Interfaces allow dynamic inspection of token-level reasoning across layers.

Expected Outcomes:
The project will produce a working explainability prototype and a human-centred evaluation of interpretability, linking technical inspection to cognitive understanding of model behaviour.

Methodology

1. **Architecture Design**: Define the functional architecture of the *Explainability Suite*, composed of three main layers: 1) **Extraction Layer** – retrieves model internals (attention weights, gradients, hidden states). 2) **Analysis Layer** – processes the raw data to compute interpretability metrics (e.g., Integrated Gradients, Attention Rollout, SHAP values). 3) **Visualization Layer** – displays results interactively via a graphical interface or notebook widgets.
2. **Model Integration**: Select one or more transformer models (e.g., BERT, RoBERTa, DistilBERT, LLaMA-2). Implement wrappers to extract attention matrices, token embeddings, and layer activations. Optionally integrate with existing libraries:
   - TransformerLens for internal state analysis.
   - Captum for gradient-based explainability.
   - Ecco for interactive transformer introspection.
3. **Interface Development**: Build an **interactive visualization dashboard** (using Streamlit, Gradio, or Plotly Dash) that allows users to: Upload text inputs and select layers or attention heads to inspect. View attention heatmaps, token influence scores, and saliency timelines. Compare attention dynamics across models or fine-tuning checkpoints.
4. **Explainability Experiments**: Evaluate which visualization methods provide *useful* insight to human observers. Conduct structured experiments: e.g., “Can users identify reasoning errors faster with visualization X or Y?” Optionally collect human feedback or usability metrics
5. **Quantitative Evaluation**: Apply correlation-based analyses (e.g., attention vs. gradient attribution) to evaluate consistency between interpretability signals. Assess stability of explanations across runs, random seeds, and model variants.

Implementation Guidelines

- Encourage modular and reusable design: each component (extraction, analysis, visualization) should be a Python module with documented APIs.
- Code organization should follow good engineering practices (OOP, testing, docstrings, reproducibility).
- Include at least one **interactive notebook demo** showing a complete explainability workflow from

input text to visualization output.

Dataset

- Any text classification or QA dataset for visual experiments.

References

- Grimsley, C., Mayfield, E., & Bursten, J. R. (2020, May). Why attention is not explanation: Surgical intervention and causal reasoning about neural models. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 1780-1790).
- Jain, S., & Wallace, B. C. (2019). Attention is not explanation. *arXiv preprint arXiv:1902.10186*.
- Raza, S., Narayanan, A., Khazaie, V. R., Vayani, A., Chettiar, M. S., Singh, A., ... & Pandya, D. (2025). Humanibench: A human-centric framework for large multimodal models evaluation. *arXiv preprint arXiv:2505.11454*.

P15. Cartographers of the Invisible

This project turns students into **semantic explorers —** "cartographers" mapping how **Large Language Models (LLMs)** represent meaning across linguistic and multimodal spaces. By combining embedding analysis, visualization, and interpretability techniques, the project aims to **make abstract representations visible**, revealing how models cluster, separate, and relate concepts. Students will design an interactive pipeline that transforms raw embeddings into **navigable semantic maps**, helping humans intuitively explore what the model "knows" and how that knowledge is structured.

Core Pipeline Sketch:
Concept embeddings are extracted from LLMs and projected into 2D/3D spaces using dimensionality reduction. Interactive visualisations are developed to explore clusters, semantic distances, and multimodal correlations.

Expected Outcomes:
Students will generate intuitive, navigable maps of semantic space, offering a visual understanding of conceptual relationships and revealing biases or clustering tendencies in model representations.

Methodology

1. **Concept and Data Selection**: Choose a set of **semantic domains** (e.g., emotions, professions, abstract concepts, or visual categories). Collect representative textual and/or visual examples for each domain (e.g., from Wikipedia, LAION captions, or ConceptNet).
2. **Embedding Extraction**: Use pretrained language or multimodal models (e.g., BERT, CLIP, LLaMA-2, or OpenCLIP) to extract embeddings for all items. Normalize embeddings and compute pairwise cosine similarity to quantify conceptual proximity.
3. **Dimensionality Reduction and Clustering**: Apply PCA, t-SNE, or UMAP to project embeddings into 2D or 3D spaces. Test different clustering algorithms (e.g., K-Means, DBSCAN, spectral clustering) to identify emergent conceptual groupings. Compare results across models or modalities (text vs. image).
4. **Visualization Interface**: Build an **interactive semantic atlas** using Python visualization

frameworks. The interface should allow users to: Explore clusters interactively and zoom into specific concept families. Search concepts and view their nearest neighbors. Compare how the same concept appears in different models or languages. Implement color coding for semantic fields (e.g., emotions, spatial terms, actions).

5. **Interpretation and Evaluation**: Analyze emergent clusters: do they align with human semantic intuition or reflect dataset biases? Examine cross-lingual or cross-modal mapping: do embeddings for the same concept converge across modalities? Optionally, compute correlation with human semantic resources (WordNet, ConceptNet).

Engineering Guidelines

- Organize the project as a modular pipeline with reusable Python components.
- Document APIs and dependencies for reproducibility.
- Optionally deploy the interactive map as a lightweight web app (Streamlit/Gradio).

Dataset

- LAION-5B — large-scale text–image dataset.
- ConceptNet — structured commonsense knowledge base for grounding concepts.

References

- Reif, E., Yuan, A., Wattenberg, M., Viegas, F. B., Coenen, A., Pearce, A., & Kim, B. (2019). Visualizing and measuring the geometry of BERT. *Advances in neural information processing systems*, *32*.
- Abnar, S., & Zuidema, W. Quantifying attention flow in transformers. arXiv 2020. *arXiv preprint arXiv:2005.00928*, *10*.
- Liang, C. X., Tian, P., Yin, C. H., Yua, Y., An-Hou, W., Ming, L., ... & Liu, M. (2024). A comprehensive survey and guide to multimodal large language models in vision-language tasks. *arXiv preprint arXiv:2411.06284*.