University of the Witwatersrand
School of Computer Science and Applied Mathematics
COMS4054A & COMS7062A & COMS7066A:
Natural Language Processing
Project

1 Instructions

1.1 Problem Setting: The Wisconsin Card Sorting Test

The Wisconsin Card Sorting Test (WCST) is a neuropsychological test used to assess a person's cognitive flexibility, specifically their ability to adapt to changing rules or task demands. In the base version of the test, participants are shown four cards with some shapes that have discernable characteristics. These characteristics are the colour of the shapes, the type of shape and the quantity of shapes. Participants are then asked to match new cards drawn from the deck to these examples cards, but are not told the sorting rule (e.g., by colour, shape, or number). They must figure it out based on feedback (correct/incorrect). After a series of correct matches, the sorting rule changes without warning, and the participant must adapt to the new rule. Performance is measured in terms of the number of correct matches, perseverative errors (repeating an old rule), and how well the participant shifts strategies. This test is used to provide insight into executive function and cognitive control, for example how well people are able to suppress their previously learning one they realise it is no longer relevant. This is highly related to how people direct their attention and use their working memory.

For the base version we will assume there are:

- four colours: $\{red, blue, green, yellow\}$
- four shapes: $\{circle, square, star, cross\}$
- four quantities: $\{1, 2, 3, 4\}$

This means there are $4 \times 4 \times 4 = 64$ unique cards in the base deck. To encode these cards we will assume an arbitrary ordering and give each card an index. This ordering will first be over colour, then shape and then quantity in the order the different values for each property are shown above. For example, index 0 is the card for $(red, circle, 1)$, index 1 is for $(red, circle, 2)$, index 4 is for $(red, square, 1)$ and index 16 is for $(blue, circle, 1)$. To encoder a trial (one round of the test) five cards will be drawn where the first four are the category cards and the fifth is the trial card. The five cards will then be followed by a separator token (sep) which should be followed by the category index the trial card belongs to, given the category cards. There are four category tokens: $\{C1, C2, C3, C4\}$. Finally, to separate trials in the context we have the EOS token. Thus, there will be 70 tokens in total, 64 card tokens from indices 0 to 63, 4 category tokens from indices 64 to 67, and two additional "special" tokens sep and EOS at indices 68 and 69. An example of one trial is shown in Figure 1. You will be given a file `wcst.py` which is able to generate valid trials.

1.2 Steps

1. Implement a transformer architecture from scratch to perform the WCST. You may use PyTorch, any of the Jax-based neural network libraries (Flax, Haiku, etc) or Tensorflow to implement the architecture but you must code all of the operations, like self-attention, yourself.
2. Train your network on the WCST.

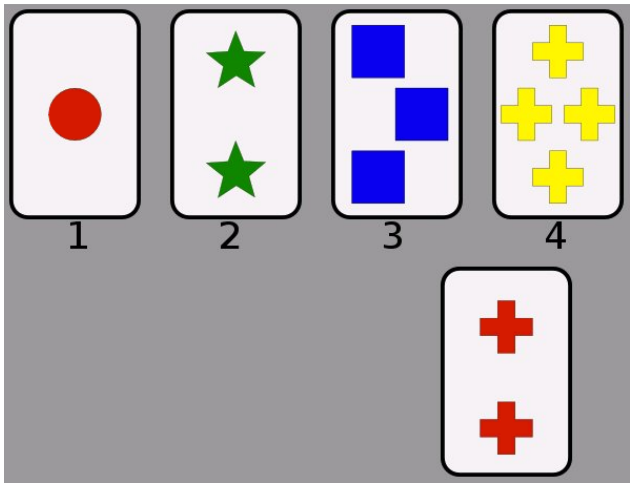

Figure 1: Example configuration of the Wisconsin Card Sorting Test: In this example the four category cards are depicted at the top of the images with their corresponding numbers below. The card to be classified is depicted at the bottom with two red crosses on it. From the image a number of classification rules could be valid. Option 1 would be correct to classify by colour, option 2 would be correct to classify by number of shapes and option 4 would be correct to match by type of shape. The agent is not told which rule to follow, but rather they must learn this from experience. However, after a certain number of trials the rule will change without the agent being alerted. Assuming the top four cards are the 1st, 41st, 22nd and 63rd cards in our deck respectively and the bottom card is the 13th, then the input to our model for this trial would be $[1, 41, 22, 63, 13, 68,]$. Following the *sep* token (remember this is given index 68 in the input) the model must perform next token prediction which will be the index for the correct category. Assuming the rule is based on colour then the model should output $[1, 41, 22, 63, 13, 68, 64]$ showing that category 1 (token $C1$) is correct. Once a trials is seen it can be kept in memory and passed in as context for a subsequent trial. All trials in the context will be separated by a EOS token (remember this is index 69). For example, the context containing two prior trials could look like: $[2, 20, 4, 43, 21, 68, 65, 69, 33, 55, 6, 9, 5, 68, 66, 69]$. If we do not substitute in the indices for the category or special tokens then the context would be: $[2, 20, 4, 43, 21, sep, C2, EOS, 33, 55, 6, 9, 5, sep, C3, EOS]$ and the input would be $[1, 41, 22, 63, 13, sep,]$ with expected output $[1, 41, 22, 63, 13, sep, C1]$.

3. Analyse the performance of the model over training. For example, you might consider the:
    - Train loss
    - Validation loss
    - Train accuracy
    - Validation accuracy
    - Final test accuracy
    - Any other metric you think is informative
4. Apply interpretation techniques to the model. For example, you might consider:
    - how the cards are embedded
    - the attention maps of the model
    - how architectural decisions affect performance
    - ideally you should try to identify the "circuits" that the model uses to solve the problem (similar to induction heads which we will discuss in class – and see Section A.2).
5. Consider how the model scales as we add more possible values for each type of card characteristic. You might consider:
    - does the model follow a clear "scaling law" (similar to the other scaling laws identified for neural language models we discuss in class – and see Section A.3)?
    - is there a tendency towards "in-context learning" when we make the dataset larger (which we will discuss in class – and see Section A.1).
6. Write a report on your findings (see below).

You may use libraries to assist you in plotting results and tracking experiments. Your mark in this project will be based heavily in terms of your research methodology and less on the overall metrics or the performance of the model. I am not certain if the model will even be able to solve the task completely anyway. You will be rewarded for noting any limitations to your approach when appropriate or identifying challenges which the model faces while training. The metrics you present will be evaluated on how appropriate they are for the question being answered. The numerical value will merely be used to determine that your model is working as best it can. Note that you must use a train-validation-test data split to train the model, optimise hyper-parameters and evaluate its performance. Data leaks will be penalised. In terms of how large your extended datasets should be - your models may benefit from getting more data but this also increases computational costs. You can use only as large a data as you need to obtain interesting results or phenomena. If you find clever or efficient ways to achieve something difficult you will be rewarded. The ideas given in the steps above are merely an indication of where you might take the project. These decisions are left deliberately vague because a large part of the assessment is seeing how you fill in these blanks yourself to understand how the model tries to solve the WCST. Finally, ensure you show your working in the write-up. If you run an experiment which doesn't work but lends some insight, present it briefly and discuss it. We will only mark you on content which you tell us in the report and so don't waste results.

You will also be expected to write a six-page (double column) report on the topic you covered, this includes figures, tables and equations. You may use unlimited space for references and can include any other information you think is relevant in supplementary material. You can direct readers to the supplement in your 6-page report, however your markers are not required to read this if they do not want to. So make sure

what you want to say is in the 6-pages. Please include your contribution statement in the appendix. The page limit is firm. You must use the IEEE format. Please see the rubric below for more details on what should be included in the write-up. This write-up should focus on your methodology, results and insight for your particular topic and should follow the format of an academic paper (provide introductions, background and so on). Creativity and insight will once again be rewarded throughout the project. Similar to your labs, you may use ChatGPT to help you write the report but as a result negative marking will be used for poor writing and formatting. Hallucinations or factually incorrect information will be heavily penalised. Your code will be checked for plagiarism and you must code the project yourselves. Finally, I will include the NeurIPS 2024 ethics statement/questionnaire as a `.tex` file on Moodle. You must append this to your report after the references but before the supplementary material and fill it out. Absence of the statement will result in the project not being marked (it takes roughly 15 minutes to fill in at the end).

2 Submission

Due Date: 28 October 2025 at 10:00.

For the submission please:

1. Submit your full code implementation (do not include `wcst.py`).
2. Submit a `requirements.txt` file with the list of packages needed and a `README.txt` file which explains the process we must follow to run your code.
3. Submit the report in `.pdf` format.

Table 1 shows the rubric which will be used to assess your write-up. I emphasise that your model accuracy or performance will only be used to determine the correctness of your approach. In other words, I care more about your thought-process, reasoning and investigation and less on your ability to obtain the best possible performance from the model. Let this guide your priorities. You will still lose method marks, however, if it is clear something was implemented wrong or there is a clear way to get better performance which was omitted (for example if hyper-parameters were not tuned at all).

3 On the Use of ChatGPT

The use of generative models to help with writing is permitted for this submission. Using generative models for coding is not permitted. Violations of this, if detected, will be dealt with as plagiarism. All other plagiarism rules of the university stand and will be implemented as usual. Finally, on Moodle there is a link to a tutorial showing how to implement a transformer from scratch. I suggest you use this tutorial as a guide, but we will be checking that you implementation is not identical. This includes copying the tutorial verbatim and then asking ChatGPT to obscure the code. Please see the course outline for a discussion about the expectations for academic integrity in this course.

References

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Aaditya K Singh, Ted Moskovitz, Felix Hill, Stephanie CY Chan, and Andrew M Saxe. What needs to go right for an induction head? a mechanistic study of in-context learning circuits and their formation. In *International Conference on Machine Learning*, pp. 45637–45662. PMLR, 2024.

A Additional Information

A.1 In-context Learning

In-context learning in large language models (LLMs) refers to the model's ability to learn and perform tasks without updating its weights, simply by being given examples or instructions within the prompt (Olsson et al., 2022). For example, if you provide a few input-output pairs (e.g., translations or math problems) in the prompt, the model can pick up on the pattern and generate the correct output for new, similar inputs. This process happens during inference, not training. It demonstrates that LLMs can temporarily adapt to new tasks using only the context provided—hence the name "in-context" learning.

A.2 Mechanistic Interpretability and Induction Heads

Mechanistic interpretability is a subfield of machine learning research that aims to understand the internal workings of neural networks—especially large language models—by identifying how specific components (like neurons or attention heads) contribute to specific computations or behaviours (Conmy et al., 2023). The goal is to move from treating models as black boxes to understanding them like we understand circuits or programs. An important discovery in this area is the role of induction heads, which are particular attention heads in transformer models that copy patterns from earlier in the context to later positions (Olsson et al., 2022; Singh et al., 2024). For instance, if a token (like a word or variable name) appears more than once, induction heads help the model reuse the earlier occurrence to predict what comes next. This mechanism underlies one way that in-context learning works: by enabling the model to recognize and generalize patterns from examples in the prompt. Induction heads effectively allow the model to "learn" from the context by mimicking the structure of the input-output examples, without modifying its weights.

A.3 Neural Scaling Laws

Neural scaling laws describe how the performance of large language models (LLMs) improves predictably as you increase model size, dataset size, and compute. Empirically, loss tends to decrease in a smooth, power-law fashion with scale—until bottlenecks like data quality or model saturation appear (Kaplan et al., 2020). This is closely related to in-context learning because research shows that larger models exhibit stronger in-context learning abilities. As models scale up, they better recognize and generalize from patterns presented in the prompt, enabling more effective few-shot or even zero-shot learning. In essence, scaling enhances the model's capacity to "learn" from context, without changing weights, by internally simulating learning algorithms.

Table 1: Rubric for the Project

Mode	0% to 20%	20% to 40%	40% to 60%	60% to 80%	80% to 100%
Write-up Structure (10%) [Negative Marking]	Adequate use of language and structure. Text in paragraphs and does not go over page limit.	Fair use of language and structure. Paragraphs, appropriate tone and within page limit.	Well written and clear. It is easy to understand the writing and one section leads naturally to the next.	Good use of language and structure. Good linking between sections, easily understood. Figures where appropriate with helpful captions.	Excellent use of language, very well written and structured. Helpful and clear figures with legible captions, labelling and legends.
Background & Related Work (10%) [Negative Marking]	No description of previous work	Some general comments but work is not self contained.	Sufficient background to make the work understandable	Background is useful and even helps contextualise the work in the field but some irrelevant information is also included	Background is sufficiently detailed to make the work understandable but also concise enough to be clear. Is very useful to contextualise the work and make the significance of the project clear.
Method (30%)	No clear description of the architecture, data or approach to training	Some description of high-level details	Fair description on the details of the base model but little elaboration of the approach to answering the sub-topic	Base approach is described in detail and steps are well justified. Adequate discussion on the approach to answering the sub-topic	Excellent description and motivation for all parts of the method including on how the sub-topic was answered
Results (30%)	Results are not given or irrelevant	Some results given with inappropriate metrics	General results presented with appropriate metrics	General and sub-topic results presented with appropriate metrics	Thorough results which are appropriate for answering the sub-topic and display the general correctness of the model
Discussion Section (20%)	No interpretation of results or incorrect interpretation. Little knowledge of the topic displayed	Some general interpretation of results broadly	Results are interpreted which display some understanding of the mechanics of the model. Displays a basic understanding of the topic	Results are interpreted and contextualized to begin to answer the sub-topic. Clear demonstration of knowledge of the general topic	Results are interpreted and contextualized to answer the sub-topic and display insight into the working of the model or original thought on the concepts