

Shortcut Detection and Mitigation via Representation Engineering

Master's Degree in Computer Science

Arianna Paolini (1943164)

Academic Year 2024/2025



SAPIENZA
UNIVERSITÀ DI ROMA



Table of Contents

1 Shortcut Learning in Large Language Models

- ▶ Shortcut Learning in Large Language Models
- ▶ Representation Engineering for Shortcut Mitigation
- ▶ Experimental Evaluation



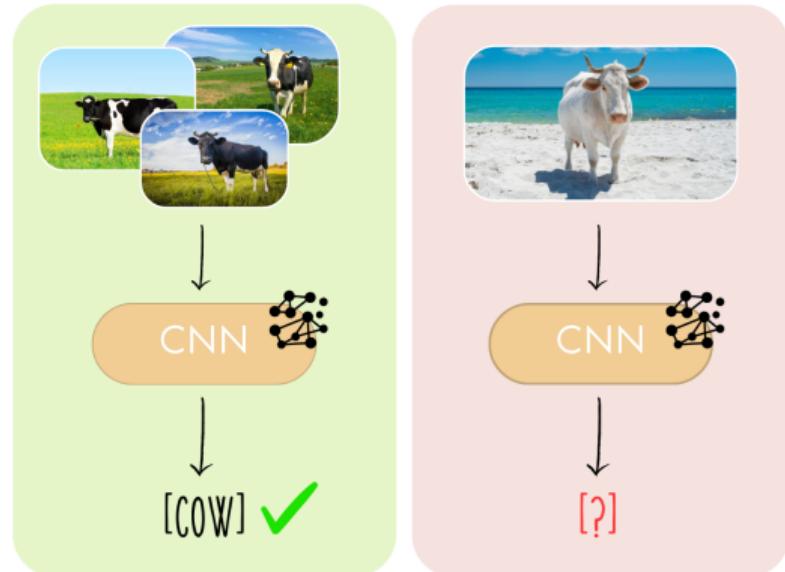
What is Shortcut Learning?

1 Shortcut Learning in Large Language Models

ML models often learn **non-robust decision rules ("shortcuts")**
e.g. *background* → *object class*

← Plausible Causes

- simplicity bias
- dataset bias





What is Shortcut Learning?

1 Shortcut Learning in Large Language Models

→ Consequences

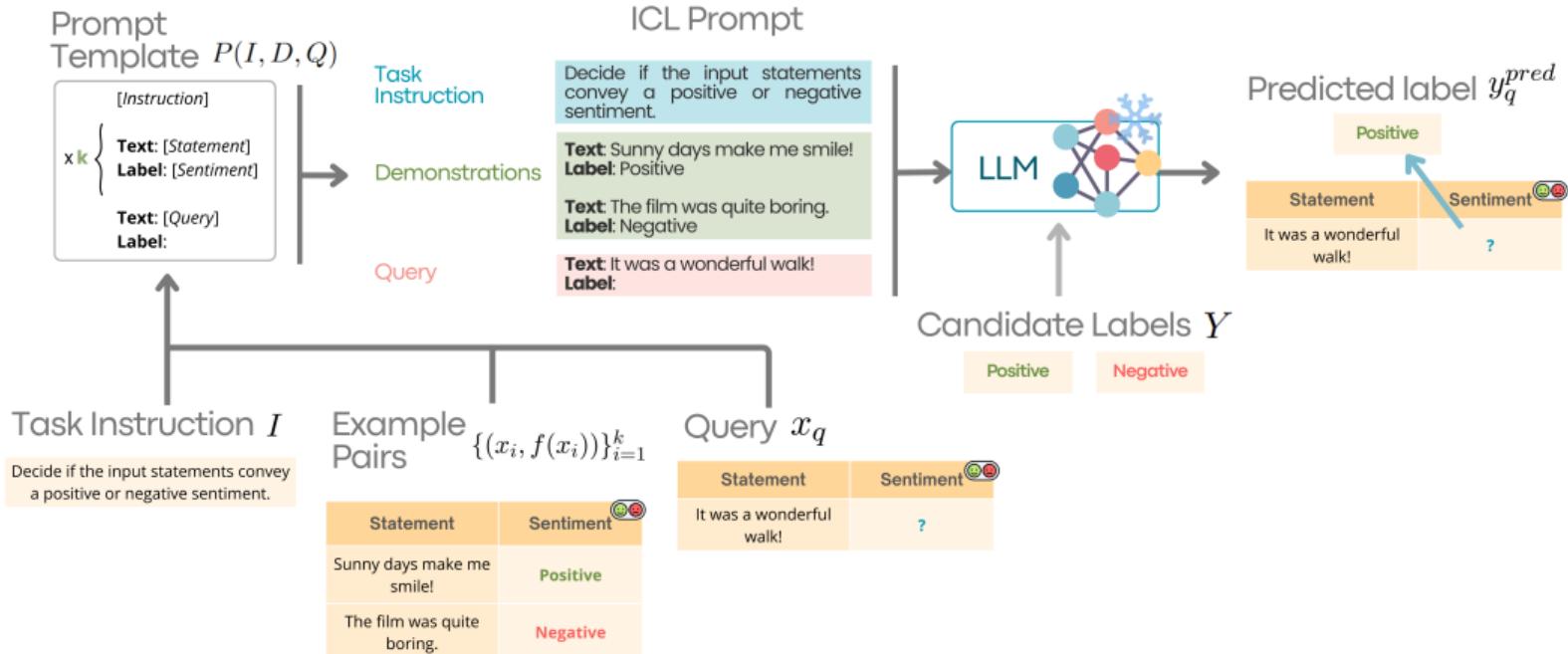
- ✓ Good performance on **training** examples and ID datasets
- ✗ Poor generalization on **OOD** data
- ✗ Undermined model interpretability





LLMs and In-Context Learning (ICL)

1 Shortcut Learning in Large Language Models





Shortcuts for LLMs under ICL

1 Shortcut Learning in Large Language Models

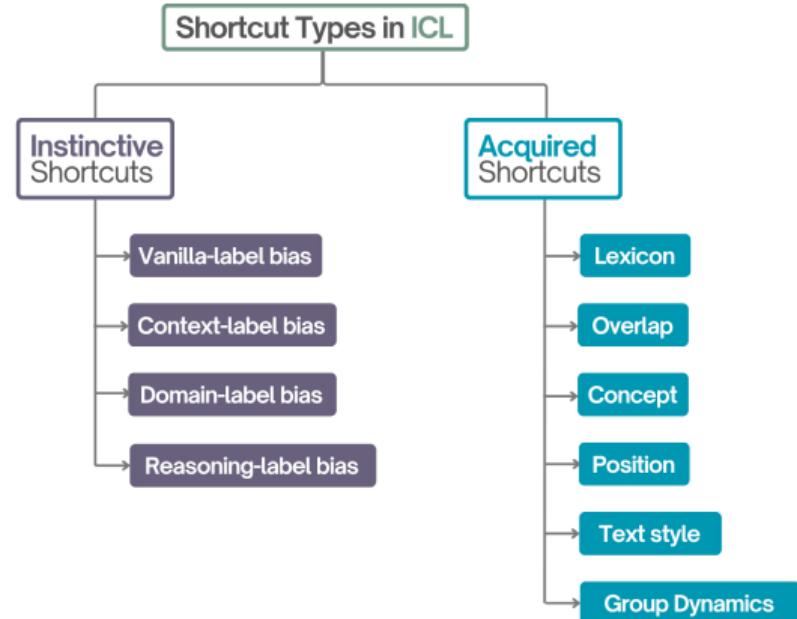
Example: Textual Entailment Recognition (TER)

Premise: Sarah has won the lottery.

Hypothesis: You will **not** believe it!

Sarah just won the lottery.

Answer: **Contradiction**





Shortcuts for LLMs under ICL

1 Shortcut Learning in Large Language Models

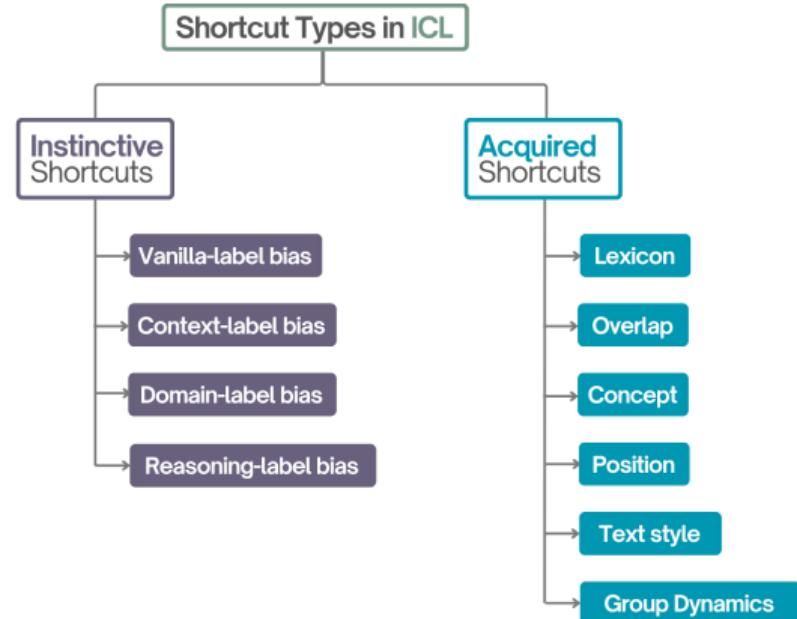
Example: Textual Entailment Recognition (TER)

Premise: Sarah has **won the lottery**.

Hypothesis: You will not believe it!

Sarah just **won the lottery**.

Answer: **Entailment**





Shortcut Detection and Mitigation

1 Shortcut Learning in Large Language Models

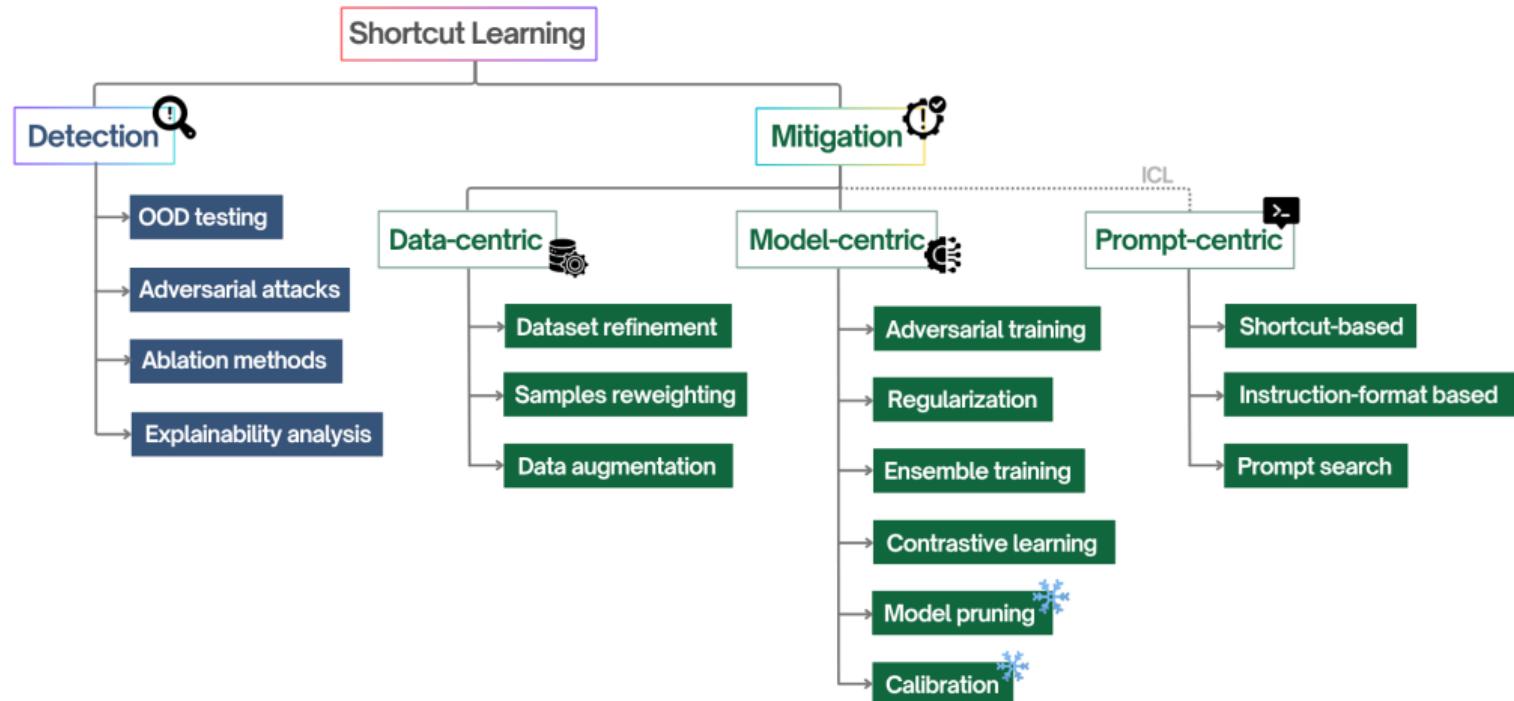




Table of Contents

2 Representation Engineering for Shortcut Mitigation

- ▶ Shortcut Learning in Large Language Models
- ▶ Representation Engineering for Shortcut Mitigation
- ▶ Experimental Evaluation



placeholder

2 Representation Engineering for Shortcut Mitigation



Beamer vs. PowerPoint

2 Representation Engineering for Shortcut Mitigation

Compared to PowerPoint, using \LaTeX is better because:

- It is not What-You-See-Is-What-You-Get, but What-You-Mean-Is-What-You-Get:
you write the content, the computer does the typesetting
- Produces a pdf: no problems with fonts, formulas, program versions
- Easier to keep consistent style, fonts, highlighting, etc.
- Math typesetting in \TeX is the best:

$$i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{r}, t) = -\frac{\hbar^2}{2m} \nabla^2 \Psi(\mathbf{r}, t) + V(\mathbf{r}) \Psi(\mathbf{r}, t)$$



Getting Started

Selecting the SINTEF Theme

To start working with `sintefbeamer`, start a \LaTeX document with the preamble:

Minimum SINTEF Beamer Document

```
\documentclass{beamer}  
\usepackage{sintef}  
\begin{document}  
\begin{frame}{Hello, world!}  
\end{frame}  
\end{document}
```



Title page

2 Representation Engineering for Shortcut Mitigation

To set a typical title page, you call some commands in the preamble:

The Commands for the Title Page

```
\title{Sample Title}  
\subtitle{Sample subtitle}  
\author{First Author, Second Author}  
\date{\today} % Can also be (ab)used for conference name &c.
```

You can then write out the title page with `\maketitle`.

To set a **background image** use the `\titlebackground` command before `\maketitle`; its only argument is the name (or path) of a graphic file.

If you use the **starred version** `\titlebackground*`, the image will be clipped to a split view on the right side of the title slide.



Writing a Simple Slide

It's really easy!

- A typical slide has bulleted lists



Writing a Simple Slide

It's really easy!

- A typical slide has bulleted lists
- These can be uncovered in sequence



Writing a Simple Slide

It's really easy!

- A typical slide has bulleted lists
- These can be uncovered in sequence

Code for a Page with an Itemised List

```
\begin{frame}{Writing a Simple Slide}
\framesubtitle{It's really easy!}
\begin{itemize}[<+>]
\item A typical slide has bulleted lists
\item These can be uncovered in sequence
\end{itemize}\end{frame}
```



Changing Slide Style

2 Representation Engineering for Shortcut Mitigation

- You can select the white or *maincolor slide style* in the preamble with `\themecolor{white}` (default) or `\themecolor{main}`
 - You should *not* change these within the document: Beamer does not like it
 - If you *really* must, you may have to add `\usebeamercolor[fg]{normal text}` in the slide
- You can change the **footline colour** with `\footlinecolor{color}`
 - Place the command *before* a new frame
 - There are four “official” colors:  maincolor,  sintefyellow,  sintefgreen,  sintefdarkgreen
 - Default is no footline; you can restore it with `\footlinecolor{}`
 - Others may work, but no guarantees!
 - Should *not* be used with the `maincolor` theme!



Blocks

2 Representation Engineering for Shortcut Mitigation

Standard Blocks

These have a color coordinated with the footline (and grey in the blue theme)

```
\begin{block}{title}  
content...  
\end{block}
```

Colour Blocks

Similar to the ones on the left, but you pick the colour. Text will be white by default, but you may set it with an optional argument.

```
\begin{colorblock}[black]{sinteflightgreen}{title}  
content...  
\end{colorblock}
```

The “official” colours of colour blocks are:  sinteflilla,  maincolor,  sintefdarkgreen, and  sintefyellow.



Using Colours

2 Representation Engineering for Shortcut Mitigation

- You can use colours with the `\textcolor{<color name>}{text}` command
- The colours are defined in the `sintefcolor` package:
 - Primary colours: `\maincolor` and its sidekick `\sintefgrey`
 - Three shades of green: `\sinteflightgreen`, `\sintefgreen`,
`\sintefdarkgreen`
 - Additional colours: `\sintefyellow`, `\sintefred`, `\sinteflilla`
 - These may be shaded—see the `sintefcolor` documentation or the [SINTEF profile manual](#)
- Do not abuse colours: `\emph{}` is usually enough
- Use `\alert{}` to bring the focus somewhere



Using Colours

2 Representation Engineering for Shortcut Mitigation

- You can use colours with the `\textcolor{<color name>}{text}` command
- The colours are defined in the `sintefcolor` package:
 - Primary colours:  `maincolor` and its sidekick  `sintefgrey`
 - Three shades of green:  `sinteflightgreen`,  `sintefgreen`,
 `sintefdarkgreen`
 - Additional colours:  `sintefyellow`,  `sintefred`,  `sinteflilla`
 - These may be shaded—see the `sintefcolor` documentation or the [SINTEF profile manual](#)
- Do not abuse colours: `\emph{}` is usually enough
- Use `\alert{}` to bring the focus somewhere
- If you highlight too much, you don't highlight at all!



Adding images

2 Representation Engineering for Shortcut Mitigation

Adding images works like in normal \LaTeX :

Code for Adding Images

```
\usepackage{graphicx}  
% ...  
\includegraphics[width=\textwidth]  
{assets/logo_RGB}
```





Splitting in Columns

2 Representation Engineering for Shortcut Mitigation

Splitting the page is easy and common; typically, one side has a picture and the other text:

This is the first column

And this the second

Column Code

```
\begin{columns}
    \begin{column}{0.6\textwidth}
        This is the first column
    \end{column}
    \begin{column}{0.3\textwidth}
        And this the second
    \end{column}
    % There could be more!
\end{columns}
```



Special Slides

2 Representation Engineering for Shortcut
Mitigation

- Chapter slides
- Side-picture slides



SAPIENZA
UNIVERSITÀ DI ROMA



Chapter slides

2 Representation Engineering for Shortcut Mitigation

- Similar to `frames`, but with a few more options
- Opened with `\begin{chapter}[<image>]{<color>}{<title>}`
- Image is optional, colour and title are mandatory
- There are seven “official” colours:  `maincolor`,  `sintefdarkgreen`,
 `sintefgreen`,  `sinteflightgreen`,  `sintefred`,
 `sintefyellow`,  `sinteflilla`.
 - Strangely enough, these are *more* than the official colours for the footline.
 - It may still be a nice touch to change the footline of following slides to the same color of a chapter slide. Your choice.
- Otherwise, `chapter` behaves just like `frame`.



Fonts

2 Representation Engineering for Shortcut Mitigation

- The paramount task of fonts is being readable
- There are good ones...
 - Use serif fonts only with high-definition projectors
 - Use sans-serif fonts otherwise (or if you simply prefer them)
- ... and not so good ones:
 - Never use monospace for normal text
 - Gothic, calligraphic or weird fonts should always be avoided



Look

2 Representation Engineering for Shortcut Mitigation

- To insert a final slide with the title and final thanks, use \backmatter.
 - The title also appears in footlines along with the author name, you can change this text with \footlinepayoff
 - You can remove the title from the final slide with \backmatter[notitle]
- The aspect ratio defaults to 16:9, and you should not change it to 4:3 for old projectors as it is inherently impossible to perfectly convert a 16:9 presentation to 4:3 one; spacings *will* break
 - The aspectratio argument to the beamer class is overridden by the SINTEF theme
 - If you *really* know what you are doing, check the package code and look for the geometry class.



Table of Contents

3 Experimental Evaluation

- ▶ Shortcut Learning in Large Language Models
- ▶ Representation Engineering for Shortcut Mitigation
- ▶ Experimental Evaluation



Good Luck!

3 Experimental Evaluation

- Enough for an introduction! You should know enough by now
- If you have any suggestions or corrections, feel free to contribute on the [GitHub repository](#)! You can [open an issue](#) or [fork the project](#) and directly propose your changes with a Pull Request.



Shortcut Detection and Mitigation via Representation Engineering

Thank you for listening!

Any questions?