PhD Thesis

Koustuv Sinha

Acknowledgements

Abstract

Abstract in French

Contributions to Original Knowledge

Contributions of Authors

List of Figures

3.1 Dataset generation pipeline. 5
3.2 Illustration of how a set of facts can split and combined in various ways across sentences. 5
3.3 Noise generation procedures of CLUTRR. 5
3.4 Systematic generalization when train on k=2 and 3. 6

4.1 Graphical representation of the Permutation Acceptance class of metrics. 8
4.2 Comparison of ω_{max}, ω_{rand}, $\mathcal{P}^c$ and $\mathcal{P}^f$ with the model accuracy $\mathcal{A}$ on multiple datasets, where all models are trained on the MNLI corpus [1]. 8
4.3 Average entropy of model confidences on permutations.. 9
4.4 BLEU-2 score versus acceptability of permuted sentences across all test datasets. 10
4.5 POS Tag Mini-Tree overlap score and percentage of permutations which the models assigned the gold label. 11
4.6 ω_x threshold for all datasets with varying x and computing the percentage of examples that fall within the threshold. 11

5.1 Downstream results on scrambled pre-training. 13
5.2 GLUE and PAWS task dev set performance when finetuned on naturally and randomly ordered text, respectively, using pre-trained RoBERTa (base) models on different versions of BookWiki corpus. 13

5.3 Risannen Data Analysis. 14

6.1 Grammatical acceptability scores on BLiMP dataset. 16

List of Tables

Contents

1 Introduction **1**

2 Background **2**

2.1 Early methods for text representation 2

2.2 Neural Inductive bias of text representation 2

2.2.1 Feed Forward Neural Networks 2

2.2.2 Recurrent Neural Networks 2

2.2.3 Transformer Models 2

2.3 Pre-training and the advent of Large Language Models 2

2.4 Systematicity and Generalization 3

2.4.1 Definitions 3

2.4.2 Tasks 3

3 Understanding semantic generalization through productivity **4**

3.1 Technical Background 4

3.2 CLUTRR: A Diagnostic Benchmark for Inductive Reasoning in Text . . . 4

3.2.1 Dataset construction 6

3.2.2 Productivity and reasoning 6

3.3 Results 6

3.4 Related Work 6

3.5 Discussion . . . 6
3.6 Follow-up findings in the community . . . 6

4 Quantifying syntactic generalization using word order 7
4.1 Technical Background . . . 7
4.2 Word Order in Natural Language Inference . . . 7
4.2.1 Probe Construction . . . 7
4.3 Experiments & Results . . . 7
4.4 Related Work . . . 7
4.5 Discussion . . . 7
4.6 Follow-up findings in the community . . . 7

5 Probing syntax understanding through distributional hypothesis 12
5.1 Technical Background . . . 14
5.2 Dataset construction and pre-training . . . 14
5.3 Experiments . . . 14
5.3.1 Downstream reasoning tasks . . . 14
5.3.2 Evaluating the effectiveness of probing syntax . . . 14
5.4 Related Work . . . 14
5.5 Discussion . . . 14
5.6 Follow-up findings in the community . . . 14

6 Measuring systematic generalization by exploiting absolute positions 15
6.1 Technical Background . . . 15
6.2 Systematic understanding of absolute position embeddings . . . 15
6.3 Related Work . . . 15
6.4 Experiments . . . 15
6.5 Discussion . . . 15

7 Conclusion **17**

7.1 Summary 17

7.2 Limitations 17

7.3 Future Work 17

8 Bibliography **18**

Bibliography **19**

Glossary 19

Acronyms 19

9 Appendix **21**

9.1 Org mode auto save 21

9.2 Remove “parts” from report 21

9.3 Add newpage before a heading 22

9.4 Glossary and Acronym build using Latexmk 22

9.5 Citation style buffer local 23

9.6 Org latex compiler options 23

Chapter 1

Introduction

Central Theme of the thesis : Understanding systematicity in pre-trained language models through semantic and syntactic generalization.

In this thesis I discuss my work on understanding systematicity in pre-trained language models.

Chapter 2

Background

2.1 Early methods for text representation

2.2 Neural Inductive bias of text representation

2.2.1 Feed Forward Neural Networks

2.2.2 Recurrent Neural Networks

2.2.3 Transformer Models

Large Language Models (LLMs) are the state-of-the-art in language models, which are based on Transformers.

2.3 Pre-training and the advent of Large Language Models

Success of pre-training and scale

2.4 Systematicity and Generalization

2.4.1 Definitions

1. Productivity

2. Word Order Sensitivity

2.4.2 Tasks

Chapter 3

Understanding semantic generalization through productivity

3.1 Technical Background

3.2 CLUTRR: A Diagnostic Benchmark for Inductive Reasoning in Text

Paper: [2]

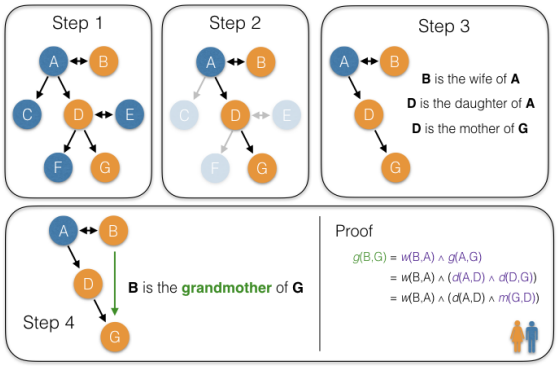

Figure 3.1 Dataset generation pipeline.

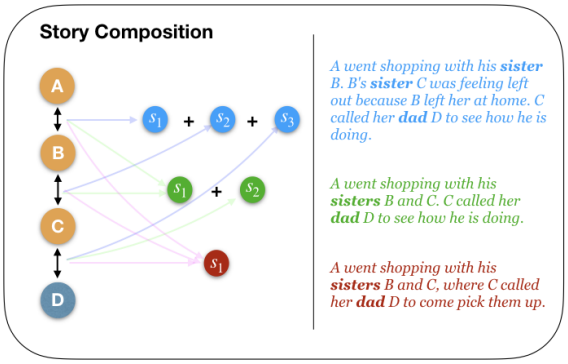

Figure 3.2 Illustration of how a set of facts can split and combined in various ways across sentences.

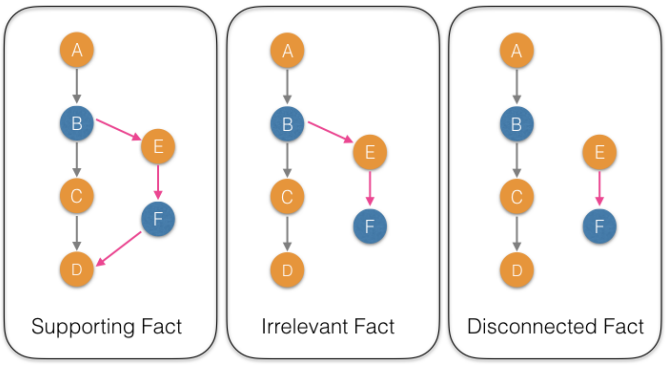

Figure 3.3 Noise generation procedures of CLUTRR.

3.2.1 Dataset construction

3.2.2 Productivity and reasoning

3.3 Results

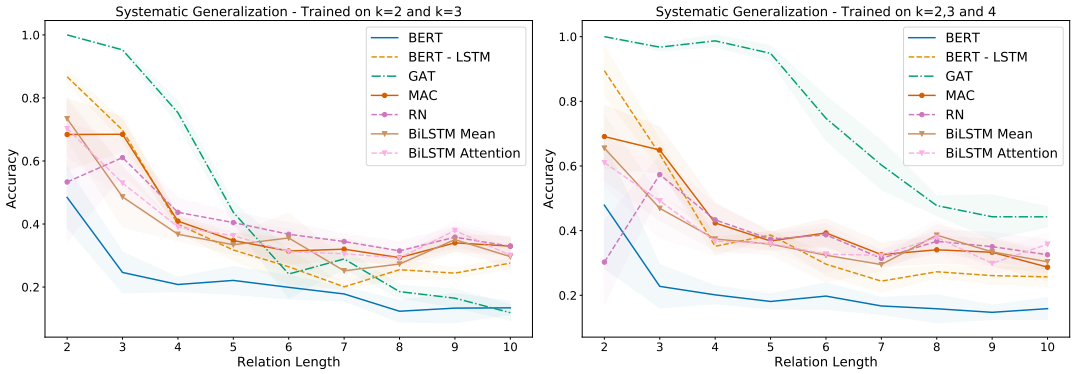

Figure 3.4 Systematic generalization when train on k=2 and 3.

3.4 Related Work

3.5 Discussion

3.6 Follow-up findings in the community

Chapter 4

Quantifying syntactic generalization using word order

Paper [3]

4.1 Technical Background

4.2 Word Order in Natural Language Inference

4.2.1 Probe Construction

4.3 Experiments & Results

4.4 Related Work

4.5 Discussion

4.6 Follow-up findings in the community

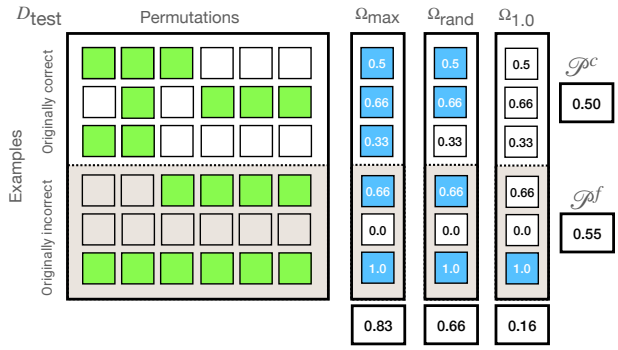

Figure 4.1 Graphical representation of the Permutation Acceptance class of metrics.

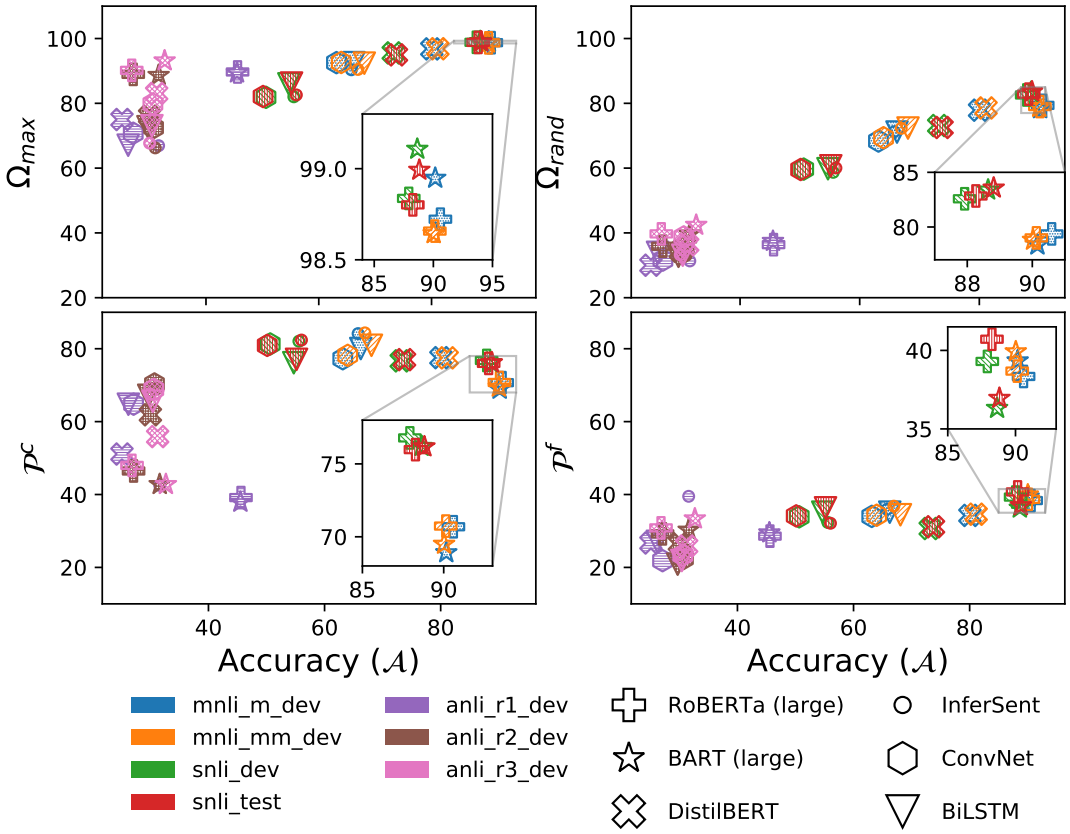

Figure 4.2 Comparison of ω_{max}, ω_{rand}, $\mathcal{P}^c$ and $\mathcal{P}^f$ with the model accuracy $\mathcal{A}$ on multiple datasets, where all models are trained on the MNLI corpus [1].

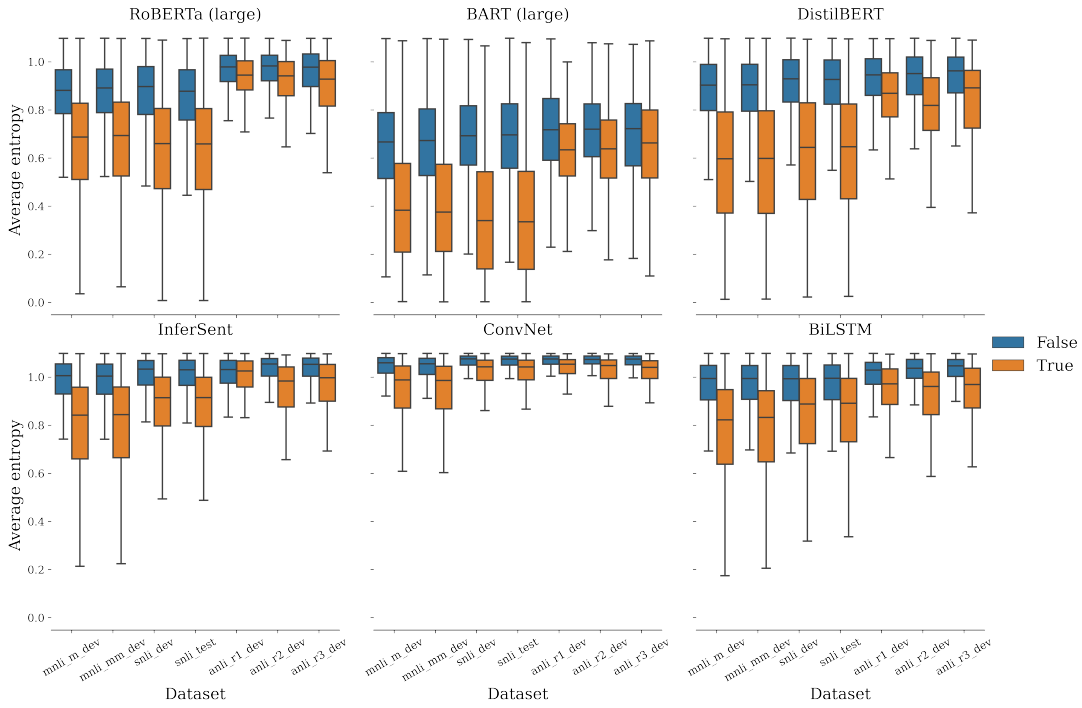

Figure 4.3 Average entropy of model confidences on permutations..

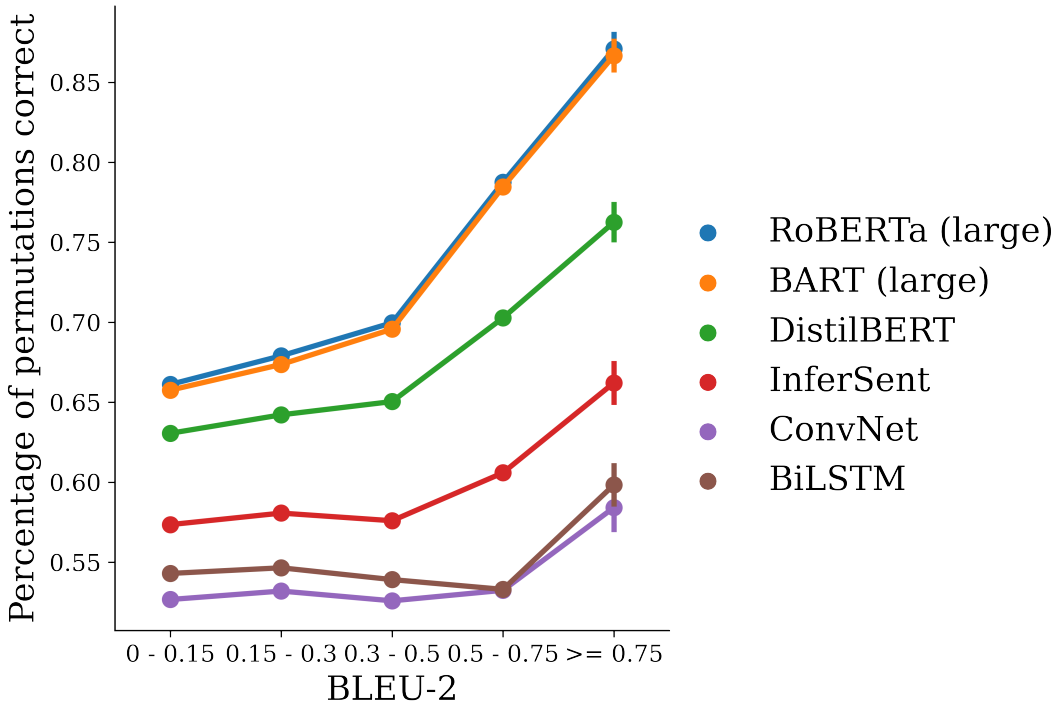

Figure 4.4 BLEU-2 score versus acceptability of permuted sentences across all test datasets.

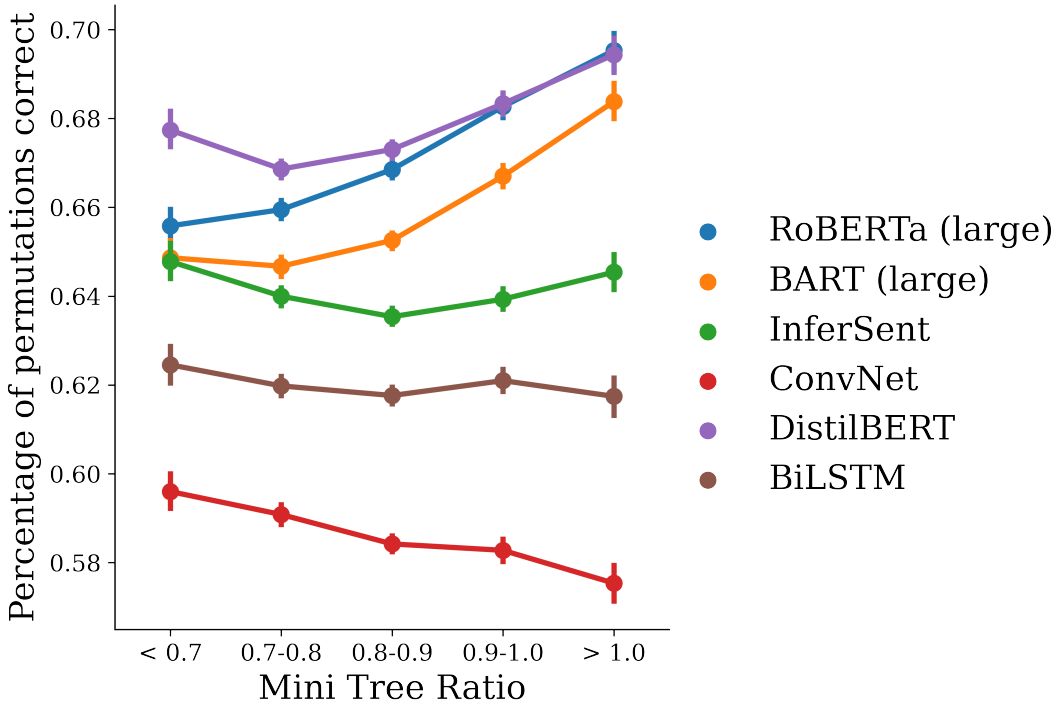

Figure 4.5 POS Tag Mini-Tree overlap score and percentage of permutations which the models assigned the gold label.

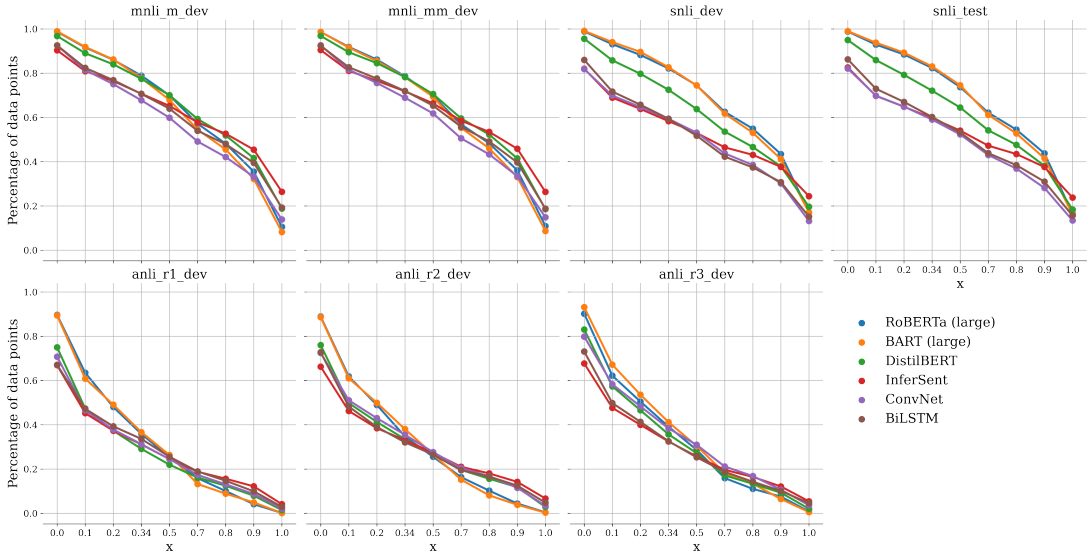

Figure 4.6 ω_x threshold for all datasets with varying x and computing the percentage of examples that fall within the threshold.

Chapter 5

Probing syntax understanding through distributional hypothesis

Paper: [4]

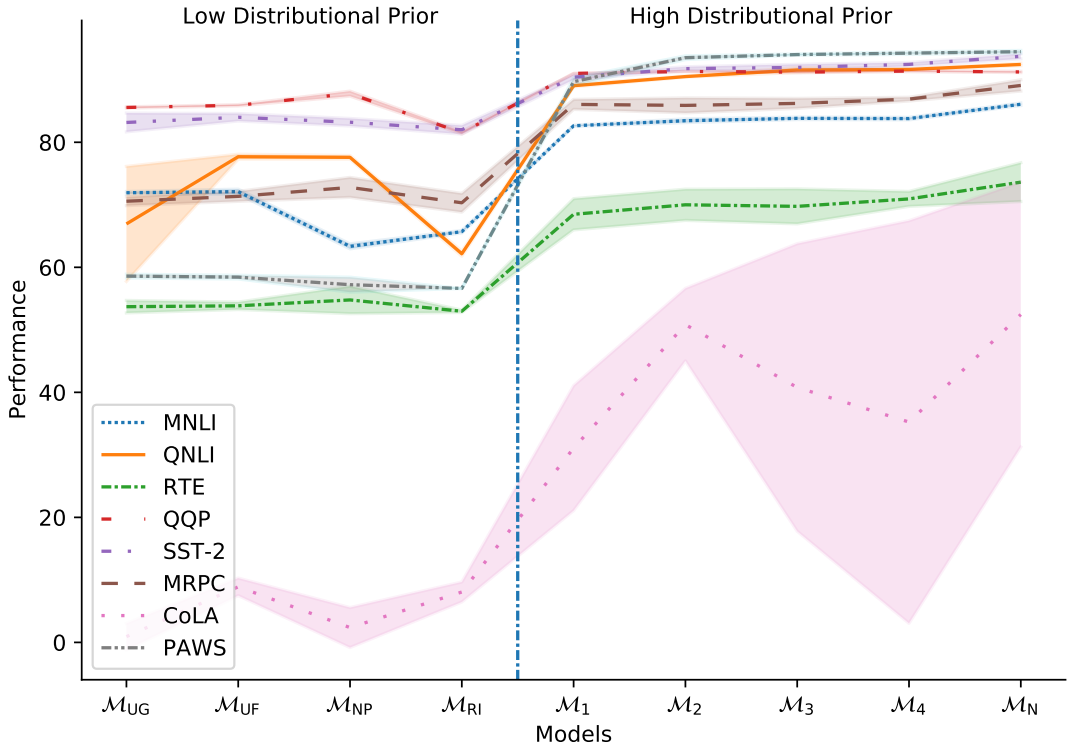

Figure 5.1 Downstream results on scrambled pre-training.

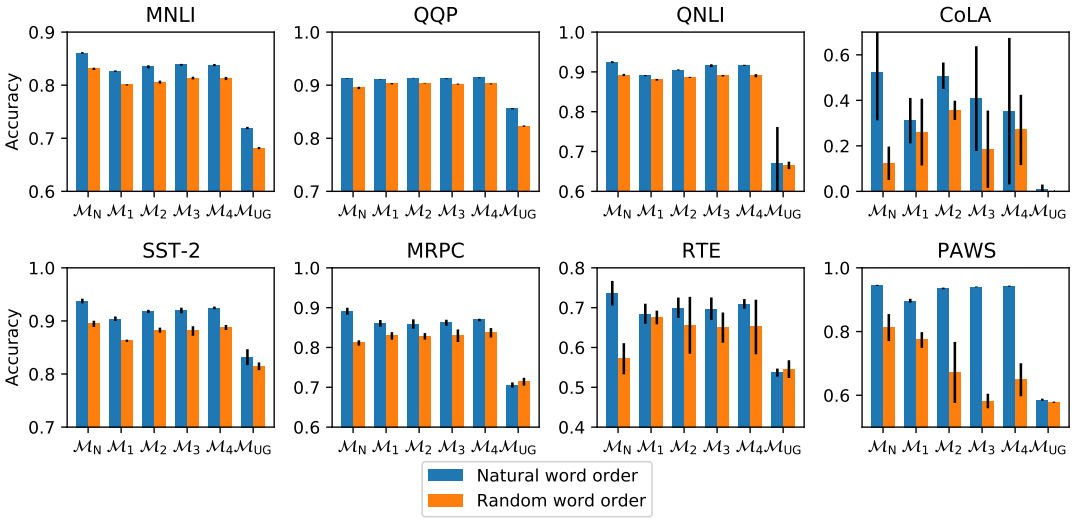

Figure 5.2 GLUE and PAWS task dev set performance when finetuned on naturally and randomly ordered text, respectively, using pre-trained RoBERTa (base) models on different versions of BookWiki corpus.

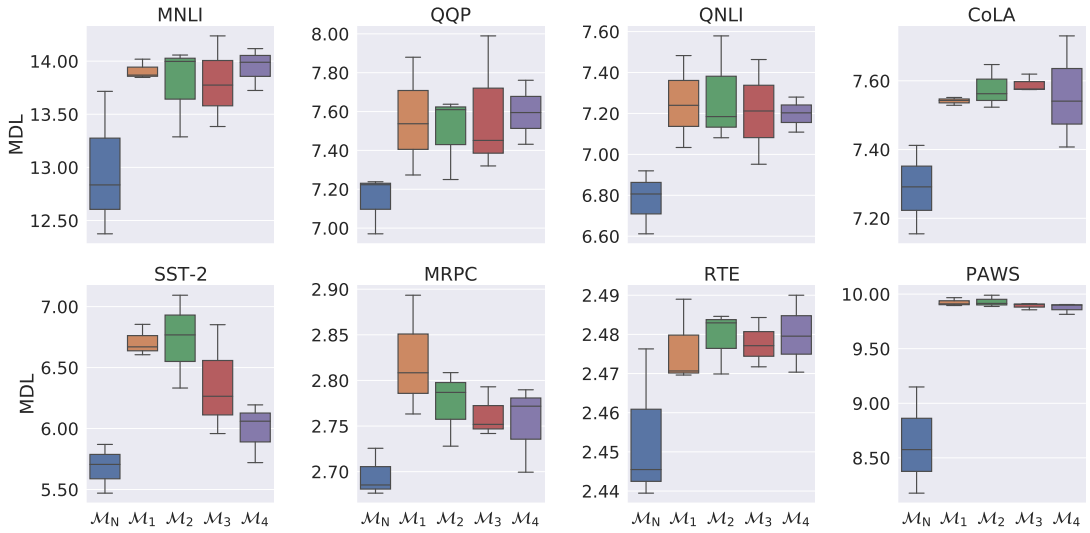

Figure 5.3 Risannen Data Analysis.

5.1 Technical Background

5.2 Dataset construction and pre-training

5.3 Experiments

5.3.1 Downstream reasoning tasks

5.3.2 Evaluating the effectiveness of probing syntax

5.4 Related Work

5.5 Discussion

5.6 Follow-up findings in the community

Chapter 6

Measuring systematic generalization by exploiting absolute positions

6.1 Technical Background

6.2 Systematic understanding of absolute position embeddings

6.3 Related Work

6.4 Experiments

6.5 Discussion

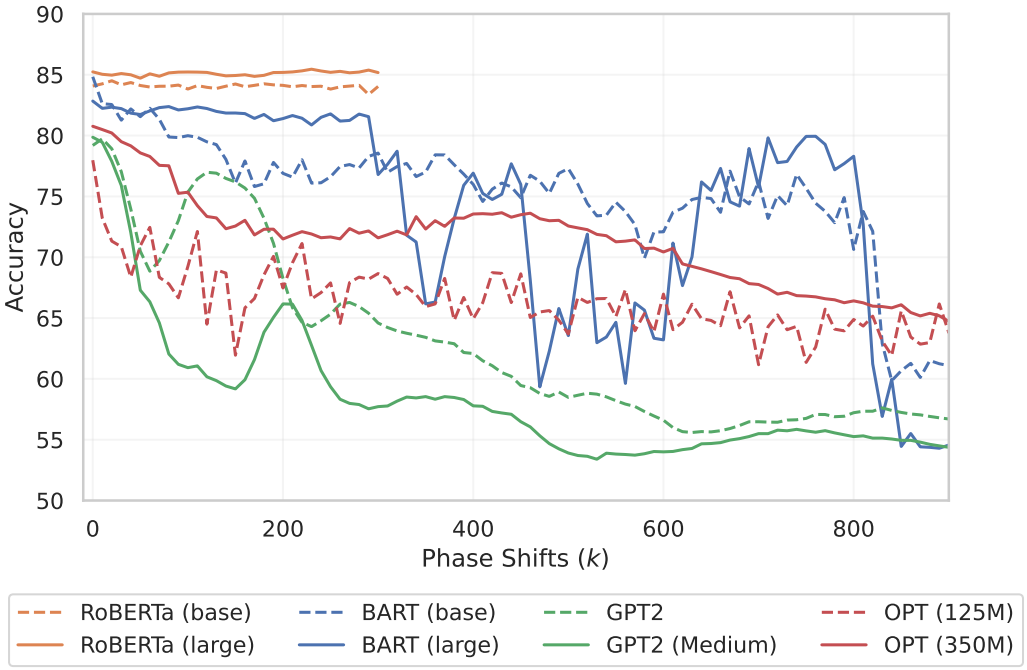

Figure 6.1 Grammatical acceptability scores on BLiMP dataset.

Chapter 7

Conclusion

7.1 Summary

7.2 Limitations

7.3 Future Work

Chapter 8

Bibliography

Bibliography

[1] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[2] Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. CLUTRR: A Diagnostic Benchmark for Inductive Reasoning from Text. In *Empirical Methods in Natural Language Processing (EMNLP) 2019*, September 2019.

[3] Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. UnNatural Language Inference. In *Association for Computational Linguistics (ACL) 2021*, June 2021.

[4] Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little. In *Empirical Methods in Natural Language Processing (EMNLP)*, April 2021.

Glossary

Transformers A class of models first derived by Vaswani et al. 2017. 2

Acronyms

LLMs Large Language Models. 2

Chapter 9

Appendix

9.1 Org mode auto save

Run the following snippet to auto save and compile in org mode.

```
(defun kdm/org-save-and-export ()
(interactive)
(if (and (eq major-mode 'org-mode)
    (ido-local-file-exists-p (concat (file-name-sans-extension (buffer-name
  (org-latex-export-to-latex)))

(add-hook 'after-save-hook 'kdm/org-save-and-export)
```

9.2 Remove "parts" from report

```
(add-to-list 'org-latex-classes
             '("report-noparts"
               "\\documentclass[11pt]{report}"
               ("\\chapter{%s}" . "\\chapter*{%s}")
```

```
                ("\\section{%s}" . "\\section*{%s}")
                ("\\subsection{%s}" . "\\subsection*{%s}")
                ("\\subsubsection{%s}" . "\\subsubsection*{%s}")))
```

9.3 Add newpage before a heading

```
(defun org/get-headline-string-element  (headline backend info)
  (let ((prop-point (next-property-change 0 headline)))
    (if prop-point (plist-get (text-properties-at prop-point headline) :pa

(defun org/ensure-latex-clearpage (headline backend info)
  (when (org-export-derived-backend-p backend 'latex)
    (let ((elmnt (org/get-headline-string-element headline backend info)))
      (when (member "newpage" (org-element-property :tags elmnt))
        (concat "\\clearpage\n" headline)))))

(add-to-list 'org-export-filter-headline-functions
             'org/ensure-latex-clearpage)
```

9.4 Glossary and Acronym build using Latexmk

Add the following snippet in the file "~/.latexmkrc": (Source: `https://tex.stackexchange.com/a/44316`)

```
add_cus_dep('glo', 'gls', 0, 'run_makeglossaries');
add_cus_dep('acn', 'acr', 0, 'run_makeglossaries');
```

```
sub run_makeglossaries {
    my ($base_name, $path) = fileparse( $_[0] ); #handle -outdir param by s
    pushd $path; # ... cd-ing into folder first, then running makeglossarie

    if ( $silent ) {
        system "makeglossaries -q '$base_name'"; #unix
        # system "makeglossaries", "-q", "$base_name"; #windows
    }
    else {
        system "makeglossaries '$base_name'"; #unix
        # system "makeglossaries", "$base_name"; #windows
    };

    popd; # ... and cd-ing back again
}

push @generated_exts, 'glo', 'gls', 'glg';
push @generated_exts, 'acn', 'acr', 'alg';
$clean_ext .= ' %R.ist %R.xdy';
```

9.5 Citation style buffer local

```
(set (make-local-variable 'bibtex-completion-format-citation-functions)
  '((org-mode      . my/bibtex-completion-format-citation-org-default-cite
```

9.6 Org latex compiler options

```
(setq org-latex-pdf-process (list "latexmk -f -pdf -%latex -interaction=no
```

Original value

```
(setq org-latex-pdf-process (list "latexmk -f -pdf %f"))
```

Let us try Fast compile `https://gist.github.com/yig/ba124dfbc8f63762f222`.

```
(setq org-latex-pdf-process (list "latexmk-fast %f"))
```

- Doesn't seem to work from Emacs.
- I need to change the save function to only export in tex. Then, have a separate process run latexmk.
- Using the python package `when-changed` to watch the thesis.tex file for change.
- Usage:

```
when-changed thesis.tex latexmk -f -pdf -interaction=nonstopmode -output-di
```

- The pdf does not update. It seems to but not always? No it does. For some reason, compilation takes ages.
- Works with `when-changed`!