PhD Thesis

Koustuv Sinha

Acknowledgements

Abstract

Abstract in French

Contributions to Original Knowledge

Contributions of Authors

List of Figures

List of Tables

Contents

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

Natural language understanding (NLU) systems have been extremely successful at reading comprehension tasks, such as question answering (QA) and natural language inference (NLI). An array of existing datasets are available for these tasks. This includes datasets that test a system's ability to extract factual answers from text [2, 3, 4, 5, 6], as well as datasets that emphasize commonsense inference, such as entailment between sentences [7, 8].

However, there are growing concerns regarding the ability of NLU systems—and neural networks more generally—to generalize in a systematic and robust way [9, 10, 11]. For instance, recent work has highlighted the brittleness of NLU systems to adversarial examples [12], as well as the fact that NLU models tend to exploit statistical artifacts in datasets, rather than exhibiting true reasoning and generalization capabilities [13, 14]. These findings have also dovetailed with the recent dominance of large pre-trained language models, such as BERT, on NLU benchmarks [15, 16], which suggest that the primary difficulty in these datasets is incorporating the statistics of the natural language, rather than reasoning.

An important challenge is thus to develop NLU benchmarks that can precisely test a model's capability for robust and systematic generalization. Ideally, we want language understanding systems that can not only answer questions and draw inferences from text, but that can also do so in a systematic, logical, and robust way. While such reasoning capabilities are certainly required for many existing NLU tasks, most datasets combine several challenges of language understanding into one, such as co-reference/entity resolution, incorporating world knowledge, and semantic parsing—making it difficult to isolate and diagnose a model's capabilities for systematic generalization and robustness.

Inspired by the classic AI challenge of inductive logic programming [17], in this chapter I discuss my work on developing semi-synthetic benchmark designed to explicitly test an NLU model's ability for systematic and robust logical generalization [18]. Our benchmark suite—termed CLUTRR (Compositional Language Understanding and Text-based Relational Reasoning)—contains a large set of semi-synthetic stories involving hypothetical families. Given a story, the goal is to infer the relationship between two family members, whose relationship is not explicitly mentioned. To solve this task, a learning agent must extract the relationships mentioned in the text, induce the logical rules governing the kinship relationships (e.g., the transitivity of the sibling relation), and use these rules to infer the relationship between a given pair of entities. Crucially, the CLUTRR benchmark allows us to test a learning agent's ability for *systematic generalization* by testing on stories that contain unseen combinations of logical rules. CLUTRR also allows us to precisely test for the various forms of *model robustness* by adding different kinds of superfluous *noise facts* to the stories.

3.1 Technical Background

3.2 CLUTRR: A Diagnostic Benchmark for Inductive Reasoning in Text

Paper: [18]

3.2.1 Dataset construction

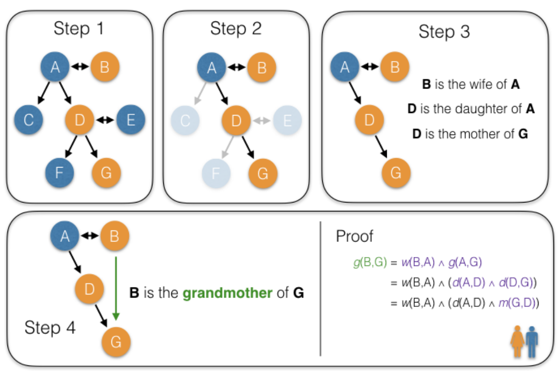

Figure 3.1 Dataset generation pipeline.

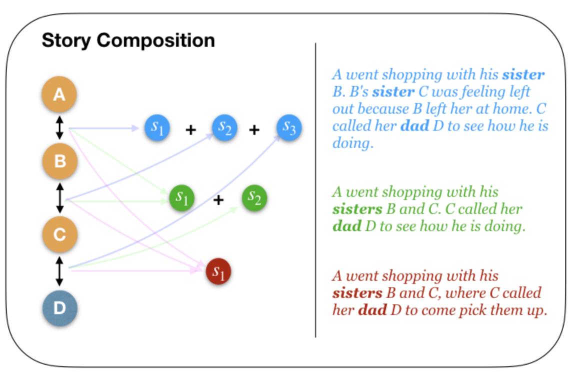

Figure 3.2 Illustration of how a set of facts can split and combined in various ways across sentences.

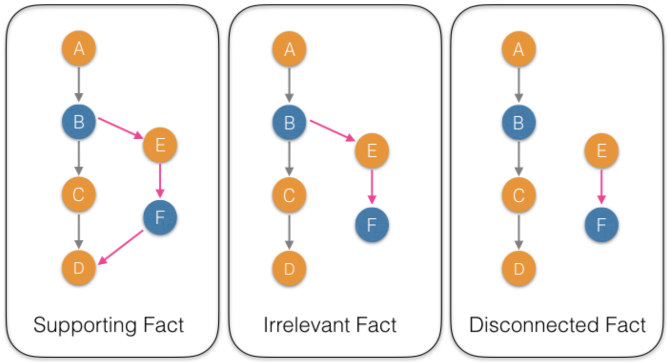

Figure 3.3 Noise generation procedures of CLUTRR.

3.2.2 Productivity and reasoning

3.3 Results

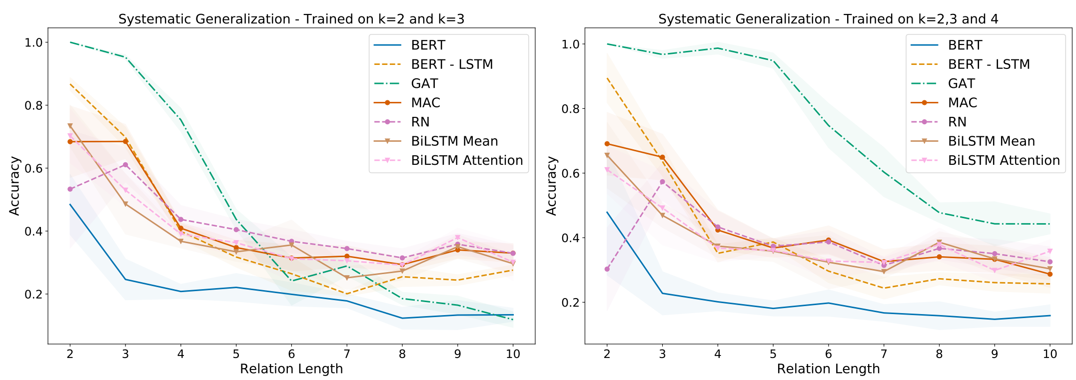

Figure 3.4 Systematic generalization when train on k=2 and 3.

3.4 Related Work

3.5 Discussion

3.6 Follow-up findings in the community

Chapter 4

Quantifying syntactic generalization using word order

Paper [19]

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

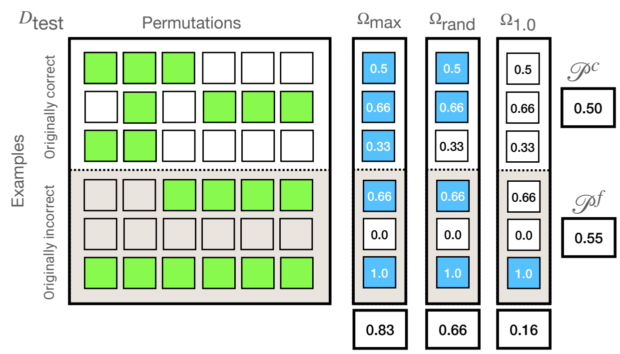

Figure 4.1 Graphical representation of the Permutation Acceptance class of metrics.

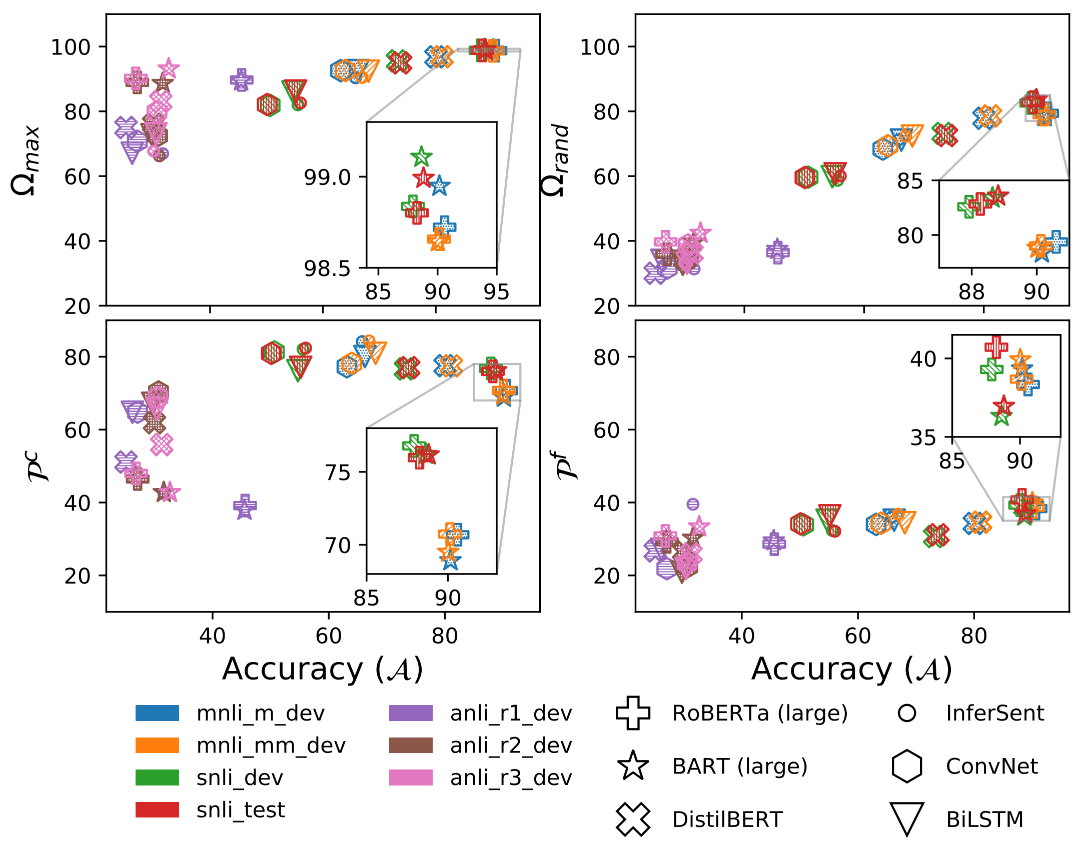

Figure 4.2 Comparison of ω_{max}, ω_{rand}, $\mathcal{P}^c$ and $\mathcal{P}^f$ with the model accuracy $\mathcal{A}$ on multiple datasets, where all models are trained on the MNLI corpus [1].

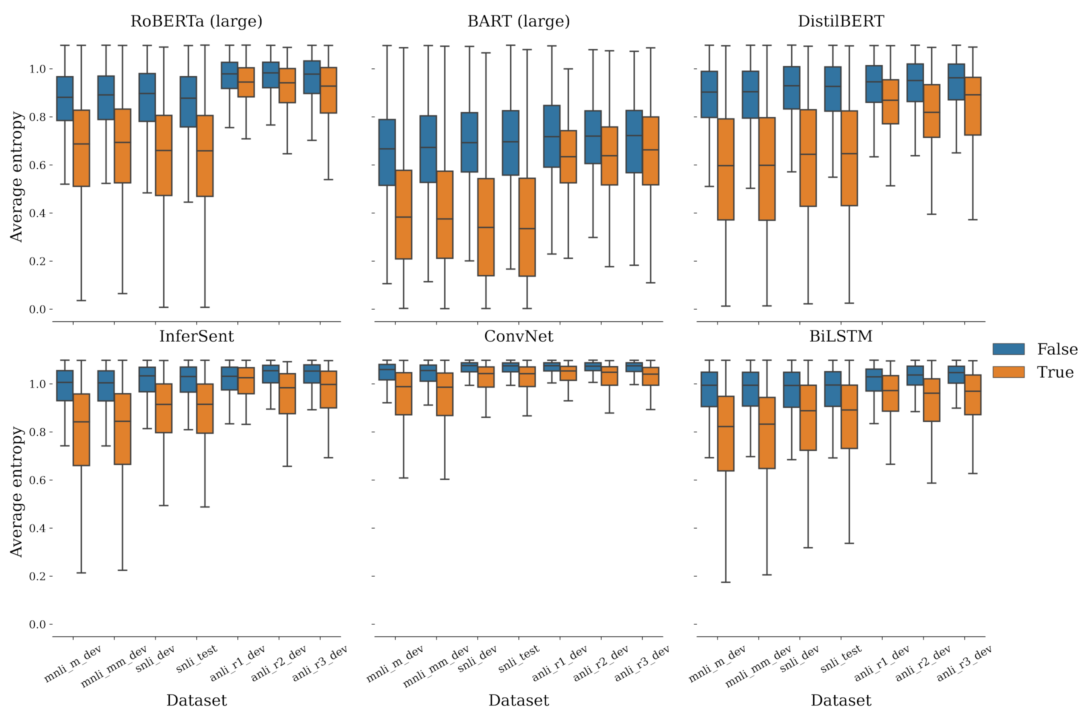

Figure 4.3 Average entropy of model confidences on permutations..

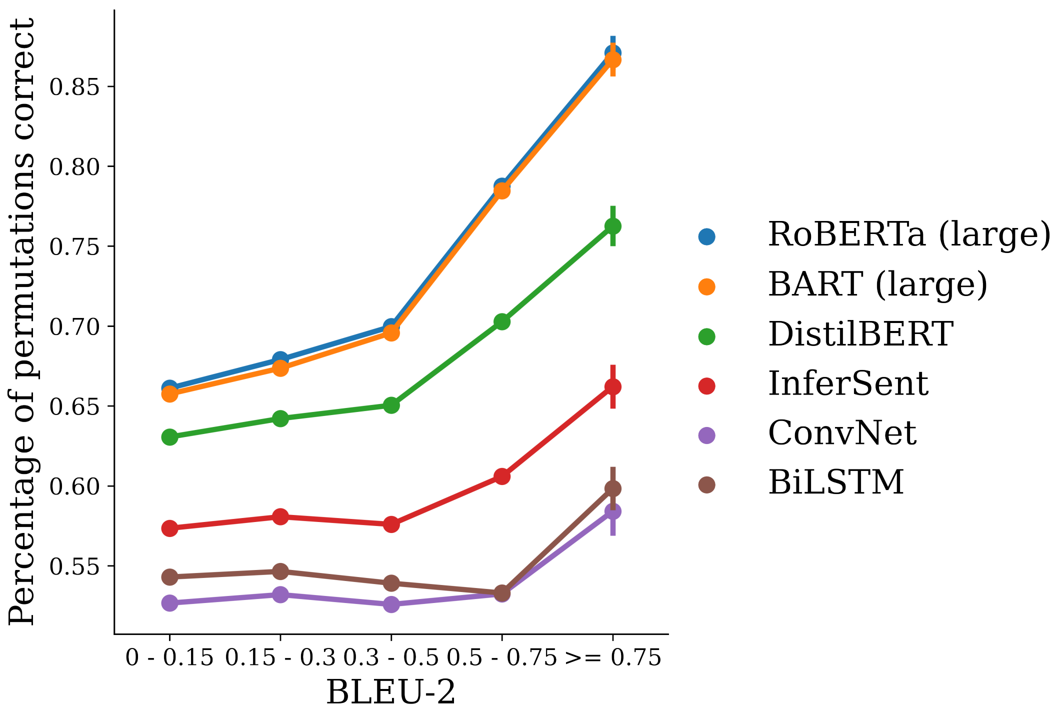

Figure 4.4 BLEU-2 score versus acceptability of permuted sentences across all test datasets.

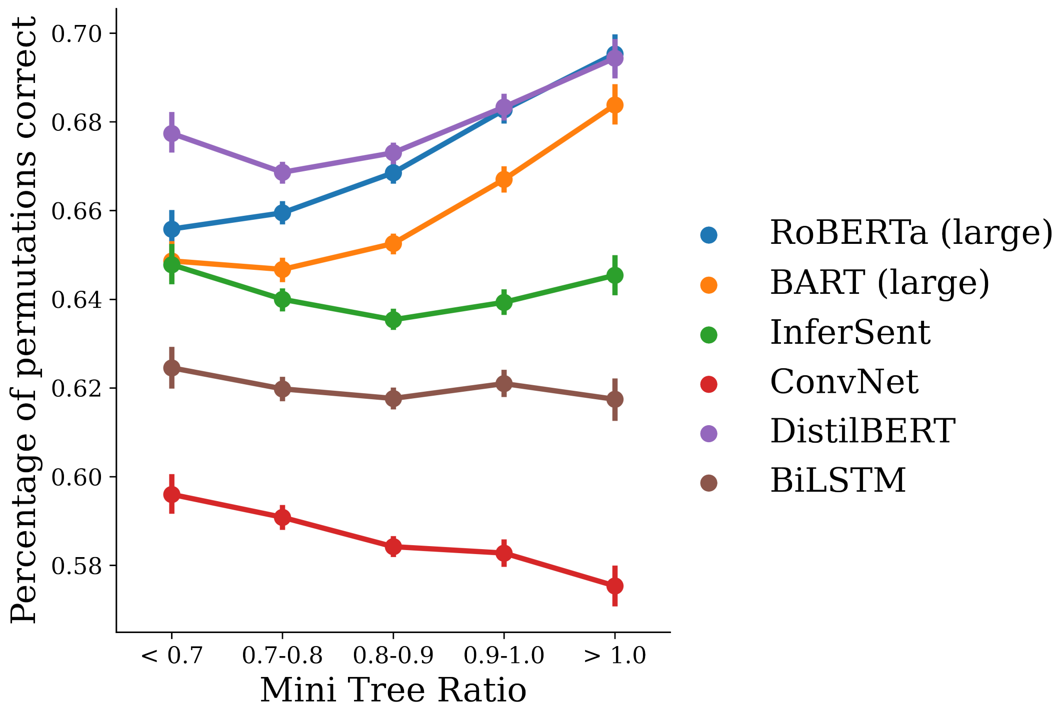

Figure 4.5 POS Tag Mini-Tree overlap score and percentage of permutations which the models assigned the gold label.

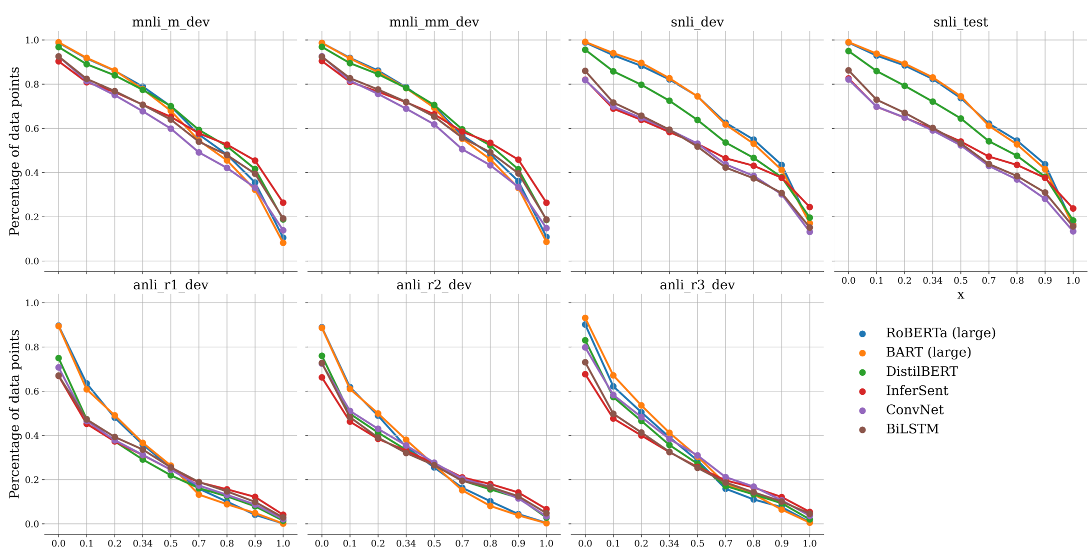

Figure 4.6 ω_x threshold for all datasets with varying x and computing the percentage of examples that fall within the threshold.

Chapter 5

Probing syntax understanding through distributional hypothesis

Paper: [20]

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

Chapter 7

Conclusion

7.1 Summary

7.2 Limitations

7.3 Future Work

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] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

[3] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[4] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. *arXiv preprint*, pages 1–12, 2016.

[5] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze

evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL-HLT*, pages 839–849, 2016.

[6] Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gur, Zenghui Yan, and Xifeng Yan. On generating characteristic-rich question sets for qa evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 562–572, 2016.

[7] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.

[8] 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)*, volume 1, pages 1112–1122, 2018.

[9] Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*, 2019.

[10] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2879–2888, 2018.

[11] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE, 2017.

[12] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, 2017.

[13] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, 2018.

[14] Divyansh Kaushik and Zachary C Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, 2018.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018.

[17] J R Quinlan. Learning logical definitions from relations. *Mach. Learn.*, 5(3):239–266, August 1990.

[18] 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.

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

[20] 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 8

Appendix

8.1 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)
```

8.2 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}")))
```

8.3 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)
```

8.4 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';
```

8.5 Citation style buffer local

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

8.6 Org latex compiler options

```
(setq org-latex-pdf-process (list "latexmk -f -pdf -%latex -interaction=nor
```

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`!