PhD Thesis

Koustuv Sinha

Contents

1 Introduction

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

2 Background

2.1 Neural Inductive bias of text representation

2.1.1 Feed Forward Neural Networks

2.1.2 Recurrent Neural Networks

2.1.3 Transformer Models

2.2 Pre-training and the advent of Large Language Models

2.2.1 Success of pre-training and scale

2.3 Systematicity and Generalization

2.3.1 Definitions

1. Productivity
2. Word Order Sensitivity

2.3.2 Tasks

3 Understanding semantic generalization through productivity

3.1 Related Work

3.2 CLUTRR: A Diagnostic Benchmark for Inductive Reasoning in Text

3.2.1 Dataset construction

3.2.2 Productivity and reasoning

3.3 Results

3.4 Discussion

3.5 Follow-up findings in the community

3.6 Summary

4 Quantifying syntactic generalization using word order

4.1 Related Work

4.2 Word Order in Natural Language Inference

4.2.1 Probe Construction

4.3 Experiments & Results

4.4 Discussion

4.5 Follow-up findings in the community

4.6 Summary

5 Probing syntax understanding through distributional hypothesis

5.1 Related Work

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 Discussion

5.5 Follow-up findings in the community

5.6 Summary

6 Measuring systematic generalization by exploiting absolute positions

6.1 Related Work

6.2 Systematic understanding of absolute position embeddings

6.3 Experiments

6.4 Discussion

6.5 Summary

7 Conclusion

7.1 Future Work

8 Bibliography

9 Acronyms

10 Appendix

10.1 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)) ".tex")))
  (org-latex-export-to-pdf)))

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

10.2 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) :parent))))

(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)
```